

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
кафедра обчислювальної техніки**

До захисту допущено:

Завідувач кафедри

_____ Сергій, СТИПЕНКО

«___» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія програмного
забезпечення комп'ютерних систем»**

спеціальності 121 «Інженерія програмного забезпечення»

**на тему: «Програмні засоби навчання системних адміністраторів
використанню можливостей технологій SDN мереж»**

Виконав (-ла):

студент (-ка) IV курсу, групи ПІ-64

Андрій ШАПРАН _____

Керівник:

Доцент, кандидат технічних наук

Олександр Долголенко _____

Консультант з нормконтролю:

Професор, доктор технічних наук

Валерій Сімоненко

Рецензент:

Доцент, кандидат технічних наук

Андрій Писаренко _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2020 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ДП 6430. 00.001 ВП	Відомість дипломного проєкту	1	
3	A4	ДП 6430. 00.002 ТЗ	Технічне завдання	3	
4	A4	ДП 6430. 00.003 ПЗ	Пояснювальна записка	51	
5	A4	ДП 6430. 00.004 Д1	Структура курсу	1	
6	A4	ДП 6430. 00.005 Д2	Спрощений блок класифікації пакетів через комктатор OpenFlow	1	
7	A4	ДП 6430. 00.006 Д3	Ідеалізований SDN контролер	1	
8	A4	ДП 6430. 00.007 Д4	Приклад опису теми курсу	10	

				ДП 6430 00.001		
	ПІБ	Підп.	Дата	Відомість дипломного проєкту	Лист	Листів
Розробн.	Шапран А.С.				1	1
Керівн.	Долголенко О.М.				КПІ ім. Ігоря Сікорського Каф. ОТ Гр. ІІІ-64	
Консульт.						
Н/контр.	Сімоненко В.П.					
Зав.каф.	Стіренко С.Г.					

Національний технічний університет України

«Київський політехнічний інститут»

Інститут (факультет) інформатики та обчислювальної техніки
(повна назва)

Кафедра обчислювальної техніки
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 121 «Інженерія програмного забезпечення»
(повна назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Сergій, СТИРЕНКО
"___" _____ 2020 року

ЗАВДАННЯ

на дипломний проєкт студента

Андрія ШАПРАНА
(ім'я, прізвище)

1. Тема проєкту «Програмні засоби навчання системних адміністраторів використанню можливостей технологій SDN мереж»

керівник проєкту к.т.н, доцент. Олександр ДОЛГОЛЕНКО,
(ім'я, прізвище, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "___" _____ 20__ року N ___

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту технічна документація, теоретичні дані, інтернет-публікації за темою роботи

4. Зміст пояснювальної записки

- Провести опис предметної області;
- Провести дослідження принципів побудови та структури SDN мереж;
- Розробити серверну платформи Moodle, що орієнтована на вивчення можливостей технологій SDN мереж;
- Використання Mininet для моделювання SDN мереж;

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	к.т.н, доцент Долголенко О. М.	10.03.2020	31.05.2020
2	к.т.н, доцент Долголенко О. М.	10.03.2020	31.05.2020
3	к.т.н, доцент Долголенко О. М.	10.03.2020	31.05.2020
4	к.т.н, доцент Долголенко О. М.	10.03.2020	31.05.2020
нормоконтроль	д.т.н., проф. Сімоненко В. П.		

7. Дата видачі завдання 15.12.2019

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту	Строк виконання етапів проекту	Примітка
1	Затвердження теми роботи	15.12.2019	
2	Вивчення та аналіз завдання	20.12.2019-02.02.2020	
3	Розробка архітектури та загальної структури системи	02.02.2020-01.03.2020	
4	Розробка модулів системи	01.03.2020-15.04.2020	
5	Програмна реалізація системи	15.04.2020-13.05.2020	
6	Оформлення пояснювальної записки	13.05.2020-19.05.2020	
8	Передзахист		
9	Захист		

Студент

(підпис)

Андрій ШАПРАН

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Олександр ДОЛГОЛЕНКО

(прізвище та ініціали)

АНОТАЦІЯ

В бакалаврській дипломній роботі реалізовано програмні засобами навчання системних адміністраторів використанню можливостей технологій SDN мереж.

Результатом роботи став навчальний курс, що дозволяє особі, що його пройде зрозуміти логіку розвитку мереж та передумови виникнення технології SDN мереж, їх сильні та слабкі сторони, принципи роботи та особливості створення і обслуговування таких мереж.

Для створення курсу була використана відкрита, та одна з найпопулярніших у світі, платформа Moodle.

ABSTRACT

In the bachelor's thesis software training systems administrators to use the capabilities of sdn network technologies.

The result was a training course that allows the person that pass it to understand the logic of network development and the prerequisites for the emergence of SDN network technology, its strengths and weaknesses, the principle of operation and features of the creation and maintenance of network.

The open and one of the world's most popular Moodle platforms was used to create the course.

Технічне завдання до дипломного проекту

ЗМІСТ

1. НАЙМЕНУВАННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ	1
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	1
3. МЕТА І ПРИЗНАЧЕННЯ.....	1
4. ДЖЕРЕЛА РОЗРОБКИ.....	1
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1. Вимоги до навчального матеріалу.....	2
5.2. Вимоги до програмної моделі серверної платформи Moodle.....	2
5.3. Вимоги до програмного забезпечення серверної платформи Moodle	2
5.4. Вимоги до апаратного забезпечення серверної платформи Moodle	2
6. ЕТАПИ РОЗРОБКИ	3

					ДП 6430. 00.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
		Шапран А. С.			Програмні засоби навчання системних адміністраторів використанню можливостей технологій SDN мереж	Літ.	Аркуш	Аркушів
Перевір.		Долголенко О.М.					1	3
Н. контр.		Сімоненко В. П.				НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64		
Затверд.								

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Назва розробки: Програмні засоби навчання системних адміністраторів використанню можливостей SDN мереж.

Область застосування: проектування моделей програмно-конфігурованих мереж.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки даного проекту стало завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр програмної інженерії», затверджене кафедрою обчислювальної техніки Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є дослідження можливостей технології SDN мереж та розробка програмних засобів навчання використанню можливостей цієї технології у віртуальному навчальному середовищі Moodle.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література, довідники, публікації в виданнях та Інтернеті, що містять опис архітектури і принципу роботи SDN та інших мереж, особливостей роботи з ними.

					ДП 6430. 00.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Програмні засоби навчання системних адміністраторів використанню можливостей технологій SDN мереж	Літ.	Аркуш	Аркушів
		Шапран А. С.					2	3
Перевір.		Долголенко О.М.				НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІП-64		
Н. контр.		Сімоненко В. П.						
Затверд.								

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до навчального матеріалу

- Достатній об'єм інформації.
- Матеріал повинен бути поданий у максимально зрозумілій для всіх формі.

5.2. Вимоги до програмної моделі серверної платформи Moodle:

- Створювати лекції, тести і завдання у вбудованому редакторі.
- Запрошувати і імпортувати користувачів, об'єднувати їх в групи, записувати їх на курси.
- Переглядати статистику активності на платформі.
- Зміна дизайну, інтеграція з іншими сервісами, візуалізація звітів.

5.3. Вимоги до програмного забезпечення серверної платформи

Moodle:

- Операційна система Microsoft Windows Server 2019
- Віртуальне навчальне середовище Moodle
- Мова сценаріїв PHP
- Працюючий сервер баз даних

5.4. Вимоги до апаратного забезпечення серверної платформи

Moodle:

- Два процесора типу Intel Xeon
- Оперативна пам'ять - 64 GB з можливістю розширення до 128 GB
- Відеоадаптер – інтегрований
- Дискові накопичувачі - 2 накопичувача не гірше, ніж SAS 8TB 7200RPM rpm

					ДП 6430. 00.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
		Шапран А. С.			Програмні засоби навчання системних адміністраторів використанню можливостей технологій SDN мереж	Літ.	Аркуш	Аркушів
Перевір.		Долголенко О.М.					3	3
						НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІП-64		
Н. контр.		Сімоненко В. П.						
Затверд.								

- Контролер SAS - не менш ніж 8 каналів з можливістю побудови RAID 0, 1, 5, 6 и 10
- Мережний контролер –1 Gb/s (2 шт. інтегровані в материнську плату)
- Серверний корпус rackmount (висота – не більше 2U)
- Два блока живлення з функцією «гарячої» заміни (1 основний + 1 резервний)
- Резервування системних вентиляторів, можливість здійснення «гарячої» заміни системних вентиляторів
- Монтажний комплект – телескопічний комплект для монтажу сервера в стойку/шафу

					ДП 6430. 00.002 ТЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
		Шапран А. С.			Програмні засоби навчання системних адміністраторів використанню можливостей технологій SDN мереж	Літ.	Аркуш	Аркушів
Перевір.		Долголенко О.М.					4	3
						НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64		
Н. контр.		Сімоненко В. П.						
Затверд.								

Пояснювальна записка
до дипломного проєкту
на тему: «Програмні засоби навчання системних
адміністраторів використанню можливостей SDN
мереж»

Київ – 2020 року

ЗМІСТ

ВСТУП	4
ОГЛЯД ТЕХНОЛОГІЇ ПОБУДОВИ SDN МЕРЕЖ ТА ІСНУЮЧИХ СИСТЕМ ЇЇ ВИВЧЕННЯ.....	6
1.1. Визначення мережі.....	6
1.2. SDN мережі.....	7
1.3. Технологія OpenFlow	9
1.3.1. OpenFlow Switch.....	10
1.3.2. OpenFlow Ports.....	13
1.4. Існуючі аналоги	14
ВИСНОВКИ ДО РОЗДІЛУ 1	18
РОЗДІЛ 2	19
2.1. Загальний обсяг інформації	19
2.2. Архітектура навчального курсу.....	19
2.3. Робота з обраною платформою.....	21
2.3.1. Установка Moodle	21
2.3.2. Перенесення курсу на серверну платформу Moodle	27
ВИСНОВКИ ДО РОЗДІЛУ 2	30
РОЗДІЛ 3	31
3.1. Розробка вступної частини.....	31
3.2. Розробка основної частини	36
3.3. Розробка завершальної частини	47
ВИСНОВКИ ДО РОЗДІЛУ 3	52

					ДП 6430. 00.003 ПЗ								
Зм.	Арк	№ докум.	Підпис	Дат									
		Шапран А. С.			Метод інтеграції та обробки мультимодальних даних ПЛІС Пояснювальна записка				Літ.	Арк	Аркушів		
		Долголенко О.М.									2	94	
									НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІП- 61				
Н. контр.		Сімоненко В.П.											

РОЗДІЛ 4	53
4.1 Доступ до курсу.....	53
4.2 Інструкція користувача.....	55
4.3 Недоліки та удосконалення	56
ВИСНОВКИ ДО РОЗДІЛУ 4	57
ВИСНОВКИ.....	58
ПЕРЕЛІК ПОСИЛАНЬ	59

					ДП 6430. 00.003 ПЗ	Арк
						3
Зм	Ар	№ докум.	Підпис	Дат		

ВСТУП

В сучасному світі доступ до мережі Інтернет має неймовірно велика кількість різноманітних пристроїв: від персональних комп'ютерів та смартфонів до розумних лампочок та виробничого обладнання. Для більшості підприємств та звичайних користувачів доступ до Інтернету та його якість має вирішальне значення. Основними пунктами при описі доступу в мережу Інтернет є стабільність, швидкість та захищеність з'єднання. Основну роль в цьому відіграє сама мережа (її принципи побудови, архітектура і т. д.). Однією з найперспективніших розробок в цьому напрямі є SDN мережі.

Мережі мають велику кількість різних реалізацій. Кожна з них має свої сильні та слабкі сторони. В залежності від розставлених пріоритетів та розмірів мережі провайдери, адміністратори мереж компаній та дата центрів використовують той чи інший підхід до організації мережі. Наприклад, для захищеного віддаленого доступу до мережі компанії часто використовується технологія VPN. Проблемою, яку SDN покликана вирішити, є гнучкість налаштування та централізоване управління мережею або її частиною. Потрібно розуміти, що SDN не потребує особливих апаратних рішень і не блокує використання уже існуючих технологій. SDN дозволяє адміністратору мережі змінити правила маршрутизації трафіку в будь-який момент. Наприклад, він може встановити трафік від одного з корпоративних клієнтів як пріоритетний, а для звичайних користувачів встановити правила, що змінять рівноправність трафіку на необхідну залежність пріоритетності та тарифу який оплачує даний клієнт чи, наприклад, за встановити голосовий трафік пріоритетним. Також SDN дає змогу централізовано збирати статистику мережі.

Загалом, сучасні технології не дозволяють так просто змінювати правила маршрутизації трафіку. У звичайному випадку необхідно змінювати правила у кожному маршрутизаторі окремо. SDN дозволяє робити всі

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		4

необхідні дії централізовано з використанням зручного графічного інтерфейсу.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		5

РОЗДІЛ 1

ОГЛЯД ТЕХНОЛОГІЇ ПОБУДОВИ SDN МЕРЕЖ ТА ІСНУЮЧИХ СИСТЕМ ЇЇ ВИВЧЕННЯ

1.1. Визначення мережі

Велика кількість сучасного програмного забезпечення має доступ до всесвітньої мережі Інтернет. Доступ до Інтернету використовується для функціонування програмного забезпечення, отримання ним актуальної інформації, оновлень і т. д.

Комп'ютерна мережа – сукупність комп'ютерів, смартфонів та інших пристроїв об'єднаних лініями зв'язку так, що вони та їх користувачі можуть обмінюватися інформацією та ресурсами (пам'ять, обчислювальні потужності і т. п.)[6].

Звісно, що для нормальної роботи мережі (особливо такої великої, як Інтернет) необхідно створити систему, що буде керувати взаємодією елементів мережі. Для цього було створено набір мережевих протоколів.

Мережевий протокол – набір правил, що створює єдиний простір обміну інформацією шляхом задання загальних правил взаємодії мережевих вузлів, систем, програмного забезпечення, визначення комп'ютерів у мережі.

Протокол задає стандарт повідомлень та способу обміну повідомленнями між комп'ютерами[6].

В зв'язку з тим, що мережі можуть бути, і є в багатьох випадках, надзвичайно складними структурами, для коректного функціонування використовується стек мережевих протоколів.

Стек протоколів – набір ієрархічно організованих мережевих протоколів, що забезпечує організацію взаємодії вузлів мережі[6].

Програмно визначена мережа (англ. software-defined networking, SDN) – одна з форм віртуалізації мережі. Мережа передачі даних, в якій рівень керування мережею реалізовано програмно та відділено від пристроїв, що здійснюють передачу інформації[3].

Важливим терміном в технології SDN, є протокол OpenFlow.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		6

OpenFlow – протокол зв'язку, який надає доступ до площини пересилання мережевого комутатора або маршрутизатора по мережі. OpenFlow використовує для зв'язку з комутатором, яким він керує, мережу, керування якою здійснюється[4].

1.2. SDN мережі

SDN є поєднанням мережевих технологій, орієнтованих на з'єднання, та технологій накладання маршрутизації, створений для подолання їх слабких сторін та збереження сильних.

SDN використовує такі ідеї:

- Для уникнення накладних витрат, що виникають при класифікації з використанням програмного забезпечення, потрібно використовувати високошвидкісне обладнання для класифікації;
- Для уникнення вузького місця, що виникає при програмній переадресації пакетів, необхідно використовувати високошвидкісне обладнання переадресації;
- Для надання менеджерам надійності та можливості інженерії трафіку, потрібно уникати використання протоколів маршрутизації для встановлення маршрутів для всього трафіку, а натомість дозволити менеджерам визначати, як обробляти кожен випадок;
- Для масштабування до розмірів Інтернету, дозвольте програмам, а не людям налаштовувати та керувати мережевими пристроями[4].

Концептуально мережевий пристрій, такий як маршрутизатор або комутатор, можна розділити на дві частини: механізми, що дозволяють менеджерам налаштовувати та керувати пристроєм та механізмами, що управляють пакетами. Для фіксації дихотомії (протипоставлення) ми використовуємо терміни площина управління та площина даних. Площина даних виконує всю обробку та пересилання пакетів. Площина управління забезпечує інтерфейс управління.

Підхід, прийнятий для SDN, відокремлює програмне забезпечення управління від базових мережевих пристроїв. Замість того, щоб повністю

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		7

переписати програмне забезпечення постачальника для управління, SDN використовує підхід розширення: програмне забезпечення SDN працює у зовнішній системі, а до пристрою додається невеликий модуль, який дозволяє зовнішній системі SDN налаштувати базове обладнання. Зовнішня система, як правило, звичайний ПК, називається контролером.

Як показано на малюнку, додавання зовнішнього контролера розширює функціональність площини управління поза пристроєм. Новий модуль SDN повинен бути доданий до площини управління пристрою, щоб дозволити зовнішньому контролеру налаштувати площину даних. На малюнку зображено модуль SDN як невеликий, оскільки код не містить розвинутої логіки, а також не забезпечує звичайний інтерфейс управління. Натомість модуль SDN просто приймає команди низького рівня від зовнішнього контролера і передає кожну команду до блоку обробки площини даних. Справа в тому, що модуль SDN мінімалістичний - він містить значно меншу складність та функціональність, ніж типовий механізм управління площиною.

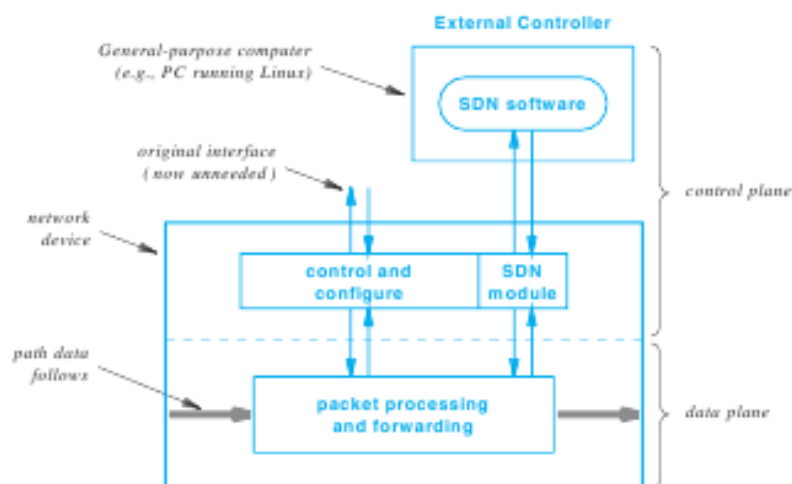


Рисунок 1.1 Концептуальна організація пристрою з новою системою управління, що замінює систему постачальника

Мабуть, найцікавіший аспект технології SDN виникає внаслідок інтеграції системи управління та мережі передачі даних. SDN використовує

той самий підхід, що і SNMP(Simple Network Management Protocol): трафік управління проходить через ті ж дроти, що і трафік даних. Тобто, замість того, щоб використовувати фізично окрему мережу, система управління використовує мережу, якою керує.

1.3. Технологія OpenFlow

OpenFlow – протокол, що використовується для зв'язку між комутатором та зовнішнім контролером[3].

Спочатку створений в Стенфордському університеті як спосіб дослідникам експериментувати з новими мережевими протоколами, OpenFlow отримав широке визнання. Багато постачальників комутаторів погодилися додати можливості OpenFlow до своїх комутаторів.

OpenFlow визначає три аспекти технології:

- Зв'язок, що використовується між контролером і комутатором;
- Набір елементів, які можна налаштувати та керувати в комутаторі;
- Формат повідомлень, які контролер і комутатор використовують для спілкування.

OpenFlow дозволяє менеджеру налаштувати переадресацію як функцію джерел, призначення та полів у заголовку пакету. Таким чином, можлива безліч конфігурацій. Кілька прикладів проілюструють деякі можливості[10]:

- Експериментальний протокол, що використовується між двома хостами;
- VLAN рівня 2, що перетинає широку область;
- Перенаправлення IP на основі джерела;
- VPN-з'єднання за запитом між двома сторонами.

OpenFlow вказує, що контролер і комутатор використовують TCP для зв'язку. Крім того, OpenFlow вказує, що спілкування має відбуватися через захищений канал зв'язку. Хоча підключення TCP дозволено, використання SSL рекомендується як спосіб гарантувати конфіденційність усієї комунікації. Важливо пам'ятати, що OpenFlow не вимагає прямого

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		9

фізичного з'єднання між контролером та мережевим пристроєм. Натомість Open-Flow передбачає, що стабільна мережа залишиться на місці і що контролер завжди зможе використовувати мережу для спілкування з мережевими пристроями, які контролюються.

Нажаль, OpenFlow не повністю відповідає меті SDN - зробити всі пристрої повністю налаштованими. Натомість OpenFlow вибирає невелику підмножину можливих пристроїв, функцій та можливостей.

OpenFlow не дозволяє менеджеру контролювати довільні пристрої. Хоча він використовувався з кількома пристроями, такими як точки доступу, основна робота в OpenFlow була спрямована на комутатори.

Спочатку версія 1.1 OpenFlow орієнтована виключно на IPv4 та пов'язані з ним протоколи підтримки, такі як ARP та ICMP. Пізніше специфікацію було розширено, щоб вона включала відповідність для IPv6 та пов'язаних з ними протоколів (наприклад, Neighbor Discovery). Зосередження уваги на IPv4 було особливо дивовижним, враховуючи первісну мету OpenFlow: стимулювати дослідження. Нові версії включають підтримку більшості протоколів, включаючи IPv6.

1.3.1. OpenFlow Switch

На рис. 1.2 зображено логічний перемикач OpenFlow, що складається з однієї або більше поточкових таблиць і групової таблиці, які виконують пошук пакетів і переадресацію, та один або кілька каналів OpenFlow до зовнішнього контролера.

					ДП 6430. 00.003 ПЗ	Арк
						10
Зм	Ар	№ докум.	Підпис	Дат		

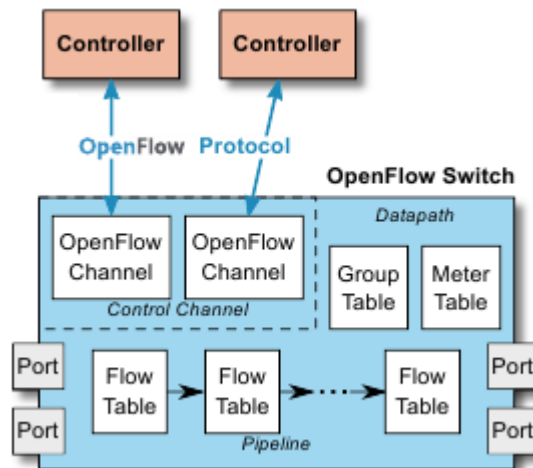


Рисунок 1.2 Основні компоненти перемикача OpenFlow

Комутатор спілкується з контролером, і контролер управляє комутатором через протокол OpenFlow. Перемикачі OpenFlow і контролери OpenFlow за замовчуванням повинні використовувати порт TCP 6653[3].

Використовуючи протокол OpenFlow, контролер може додавати, оновлювати та видаляти власні записи у власних таблицях. Кожна власна таблиця в комутаторі містить набір власних записів; кожен власний запис складається з полів збігів.

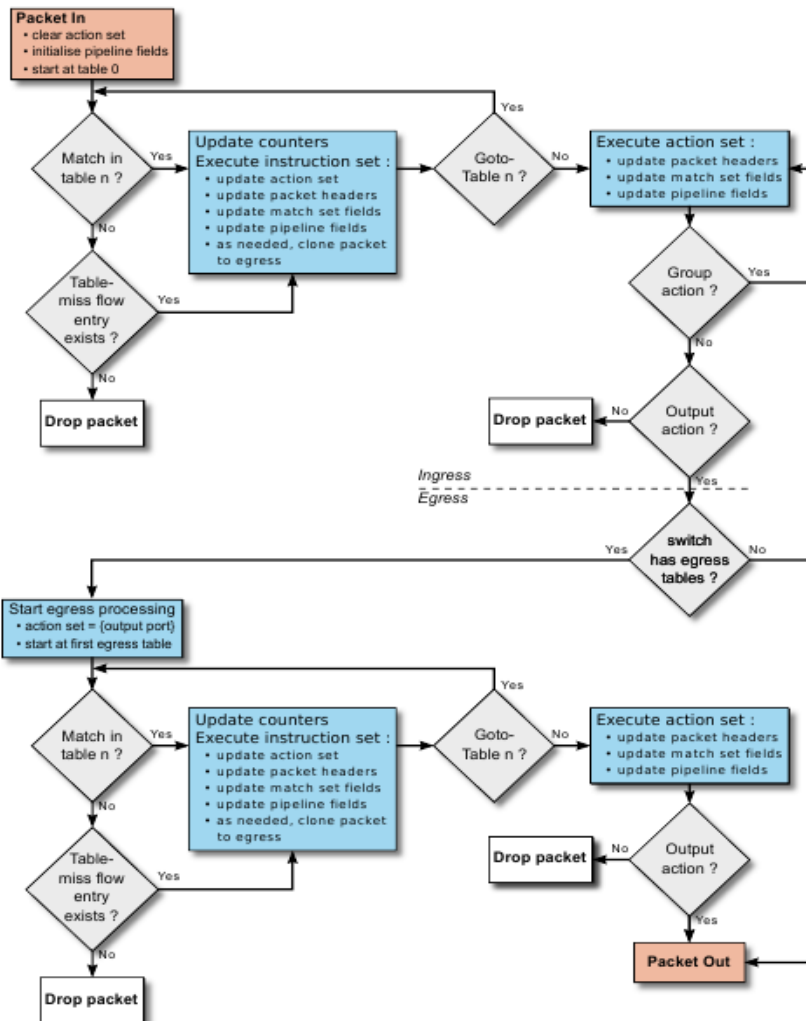


Рисунок 1.3 Спрощений блок класифікації пакетів через комутатор OpenFlow

Після отримання пакету OpenFlow Switch виконує функції, показані на рисунку 1.3.

Перемикач починає з виконання пошуку таблиці в першій таблиці, і на основі обробки конвеєра може здійснювати пошук таблиць в інших власних таблицях.

Поля заголовків пакетів витягуються з пакета, а поля конвеєра пакетів отримуються. Поля заголовків пакетів, які використовуються для пошуку таблиць, залежать від типу пакету і зазвичай включають різні поля заголовків протоколу, такі як адреса джерела Ethernet або адреса призначення IPv4[3]. На додаток до заголовків пакетів, відповідність також може бути виконана до вхідного порту, поля метаданих та інших полів

конвеєра. Метадані можуть використовуватися для передачі інформації між таблицями в комутаторі. Поля заголовків і поля конвеєра представляють пакет у його поточному стані, якщо дії, застосовані в попередній таблиці за допомогою інструкції «Застосувати дії», змінили заголовки або поля конвеєра, ці зміни відображаються в полях заголовків пакетів і полях конвеєра.

1.3.2. OpenFlow Ports

Порти OpenFlow - це мережеві інтерфейси для передачі пакетів між обробкою OpenFlow та іншою частиною мережі. Перемикачі OpenFlow логічно з'єднуються один з одним через свої порти OpenFlow, пакет може бути перенаправлений з одного перемикача OpenFlow до іншого комутатора OpenFlow лише через вихідний порт OpenFlow на першому перемикачі та вхідний порт OpenFlow на другому комутаторі[3].

Перемикач OpenFlow робить ряд портів OpenFlow доступним для обробки OpenFlow. Набір портів OpenFlow може бути не ідентичним набору мережевих інтерфейсів, наданих апаратним забезпеченням комутатора, деякі мережеві інтерфейси можуть бути відключені для OpenFlow, а перемикач OpenFlow може визначити додаткові порти OpenFlow.

Пакети OpenFlow приймаються на вхідний порт і обробляються конвеєром OpenFlow, який може пересилати їх на вихідний порт. Порт введення пакету є властивістю пакету по всьому конвеєру OpenFlow і являє собою порт OpenFlow, на якому пакет був прийнятий в комутатор OpenFlow. Вхідний порт може використовуватися для узгодження пакетів. Конвеєр OpenFlow може вирішити відправити пакет на вихідний порт, використовуючи вихідну дію, що визначає, як пакет повертається до мережі.

1.4. Існуючі аналоги

Навчальних матеріалів по темі SDN мереж є досить багато. Сюди відносяться і статті, опубліковані на різних інформаційних ресурсах і книги. Одна з найбільш відомих [4] показана на рис. 1.4.

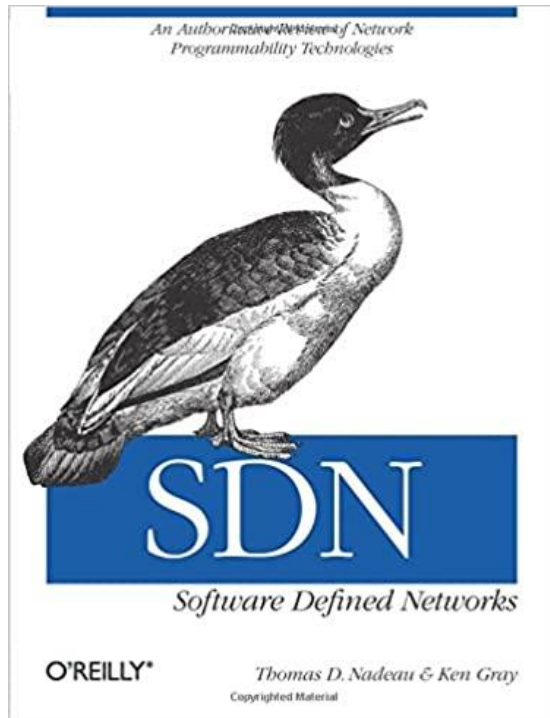


Рисунок 1.4 Книга SDN: Software Defined Networks

До плюсів даної книги можна віднести:

- Автори книги є реальними експертами в сфері;
- Чіткість у поданні інформації (інформація сформульована так, щоб не виникало подвійного трактування написаного);
- Великий обсяг поданої інформації;
- Книга дає можливість подивитися на предметну область з точки зору людей, що безпосередньо працюють з нею.

Але, нажаль, книга не є ідеальною і, навіть, не є близькою до цього звання. Велика кількість читачів, і я в їх числі, виділяють такі недоліки книги:

- Загальна непрофесійність авторів у питанні написання книги;
- Погана структурованість інформації;
- Складність розуміння поданої інформації без суттєвих знань;

					ДП 6430. 00.003 ПЗ	Арк
						14
Зм	Ар	№ докум.	Підпис	Дат		

- Погано описані причини та передумови виникнення технології SDN.

Також до недоліків можна віднести відсутність перекладу. Книга є лише у варіанті англійською мовою.

Другим очевидним джерелом інформації для вивчення теми SDN мереж є різноманітні статті та публікації[2]. Такі статті, зазвичай, покликані подати всю необхідну інформацію максимально швидко та доступно, на противагу книгам. Однак при спробі виправити недоліки книг, нажаль, втрачаються і їх переваги. Тому у більшості статей інформації достатньо лише для загального ознайомлення з темою. Також при спробі максимально зрозуміло подати інформацію часто втрачається її точність. Важливо сказати, що на багатьох інформаційних порталах, де можуть бути опубліковані статті, є коментарі, що означає, що автору статті можна задати якесь питання або уточнення, а він може відповісти і таким чином покращити свою статтю. Подібна інтерактивність однозначно є перевагою. Якщо ж ти щось не зрозумів читаючи книгу то тобі доведеться шукати цю інформацію в інших джерелах, а це, зазвичай, займає багато часу і інколи призводить до результату, що є протилежним очікуваному (замість знаходження відповідей на свої питання, питань стає більше).

Інший, з найбільш популярних навчальних матеріалів по SDN мережам[3], приведений на рис. 1.5

					ДП 6430. 00.003 ПЗ	Арк
						15
Зм	Ар	№ докум.	Підпис	Дат		

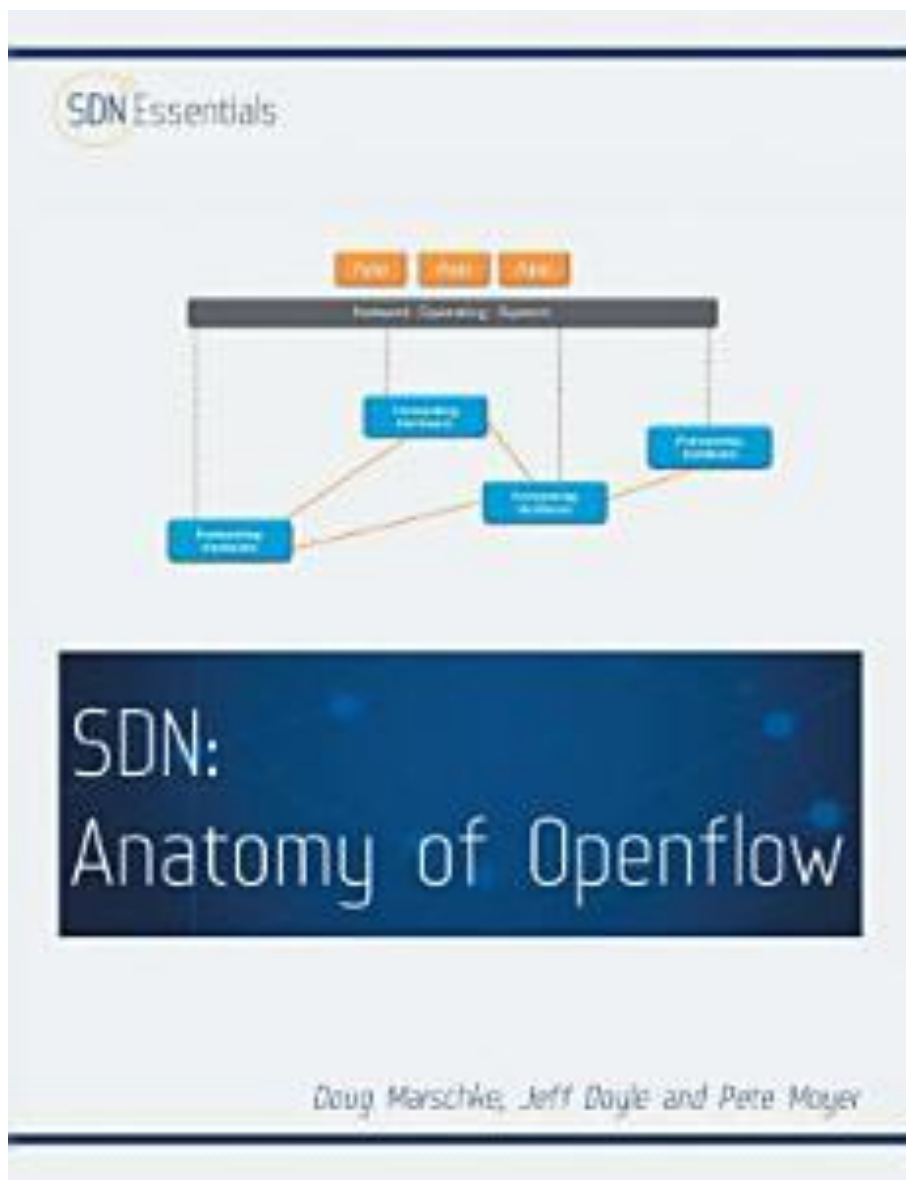


Рисунок 1.5 Книга SDN: Anatomy of Openflow

Ця книга також має свої сильні та слабкі сторони. Так до плюсів даної книги можна віднести:

- Написана так, що зацікавлює читача;
- Структуризація інформації послідовність її подання;
- Простота та зрозумілість подання інформації;
- Чіткість формулювань.

Насправді, більшість читачів дуже добре відгукуються про цю книгу, але, звісно, в ній також є недоліки:

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		16

- Це лише перша частина (перший том), а отже в книзі подана не вся необхідна інформація;

- Книгу не перекладено на інші мови, що ускладнює розуміння інформації особами, для яких англійська не є рідною.

Отже, для об'єднання всіх переваг та позбавлення можливих недоліків потрібно створити курс в якому буде достатньо великий об'єм інформації, ця інформація повинна бути добре структурована та чітко і головне зрозуміло описана. І, звісно, повинна бути інтерактивність, можливість задавати питання та отримувати на них відповіді.

					ДП 6430. 00.003 ПЗ	Арк
						17
Зм	Ар	№ докум.	Підпис	Дат		

ВИСНОВКИ ДО РОЗДІЛУ 1

В даному розділі було розглянуто особливості технології побудови SDN мереж та існуючі можливості по вивченню цієї технології. Досліджені існуючі аналоги створюваного курсу, визначено їх сильні та слабкі сторони, а також можливі шляхи вирішення їх проблем для створення найбільш ефективного курсу для вивчення SDN мереж.

					ДП 6430. 00.003 ПЗ	Арк
						18
Зм	Ар	№ докум.	Підпис	Дат		

РОЗДІЛ 2

ПРОЕКТУВАННЯ КУРСУ

2.1. Загальний обсяг інформації

Весь обсяг курсу по ефективному засвоєнню технології побудови SDN мереж можна поділити на такі пункти (теми):

- Загальний опис історії розвитку галузі;
- Опис загальної концепції (ідеї) SDN мереж;
- Опис архітектури SDN та зовнішнього контролера;
- Опис загальної концепції OpenFlow;
- Детальний розбір роботи OpenFlow;
- Детальний розбір SDN та SDN Framework;
- Використання системи Mininet для моделювання SDN мереж.

2.2. Архітектура навчального курсу

Відповідно до вище описаного поділу усього об'єму інформації, що повинна бути представлена в курсі, курс можна поділити на три класичні частини (див. рис. 2.1).

У вступі не надається ніякої інформації про технологію SDN. Вся інформація стосується лише загального устрою мережі, історії, проблем та технологій, що були покликані їх вирішити, та усього іншого, що призвело до створення SDN технології.

В основній частині знаходиться вся необхідна технічна інформація, що стосується SDN та суміжних технологій: загальний опис, концепція, архітектура, принцип роботи, технічні можливості, переваги та недоліки.

В завершальній частині відбувається опис використання технології, її потенціалу, проблем, які потребують вирішення у майбутньому, та можливих сценаріїв розвитку технології та галузі у майбутньому.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		19

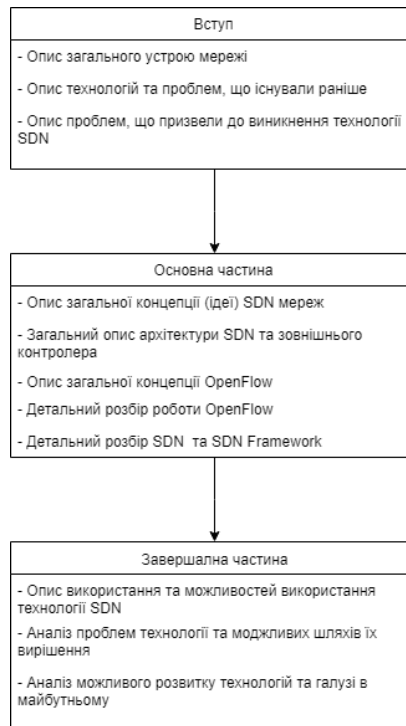


Рисунок 2.1 Архітектура навчального курсу

Інформація повинна подаватися саме в такому порядку, бо саме так з токи зору логіки повинно розвиватися та поглиблюватися уявлення та знання про тему курсу. Спочатку потрібно зрозуміти причину виникнення технології, потім її концепцію та загальний принцип роботи. Після отримання загального уявлення потрібно починати поглиблювати знання більш детальною та конкретною інформацією. Поглиблення знань відбувається саме з розгляду протоколу OpenFlow, бо саме він лежить в основі технології SDN. Тільки після огляду того, як зовнішній контролер може керувати мережевим пристроєм, можна більш детально розглядати саму технологію SDN.

Подібна структуризація інформації дозволить проводити швидко та просту навігацію по темам курсу та значно спростить вивчення та пошук необхідної інформації. Наприклад, якщо особі, що вже пройшла курс чи просто розібралася в темі до цього, потрібно знайти якусь конкретну технічну інформацію, то їй просто потрібно в основній частині курсу знайти відповідний розділ. Якщо ж, наприклад, особа зрозуміла те, як працює дана

технологія, але так і не зрозуміла навіщо вона потрібна, то їй потрібно пройти лише вступну та завершальну частини курсу.

2.3. Робота з обраною платформою

Для даного курсу було обрано платформу Moodle. Вона дозволяє для створення курсу використовувати або власний веб сервер або ж розмістити його на хостингу. Я обрав варіант в якому веб сервер буде розташовано на моєму власному ПК, бо такий варіант дозволить мені мати прямий доступ, безпосередній контроль та впевненість в тому, що я матиму доступ до серверу на час створення курсу.

2.3.1. Установка Moodle

Алгоритм установки Moodle є дуже простим. Спочатку потрібно скачати відповідний архів з офіційного сайту (див. рис. 2.2)[12].

Другим кроком є вилучення всіх файлів з архіву в підготовлену папку та запуск файлу “Start Moodle.exe” (див. рис. 2.3).

Після цього запуститься сервер Moodle, що дозволить нам перейшовши за відповідною localhost адресою в браузері, завантажити та встановити всі необхідні компоненти та провести налаштування нашого Moodle сервера.

Спочатку відкриється сторінка на якій потрібно обрати зручну мову інсталяції (див. рис. 2.4).

					ДП 6430. 00.003 ПЗ	Арк
						21
Зм	Ар	№ докум.	Підпис	Дат		

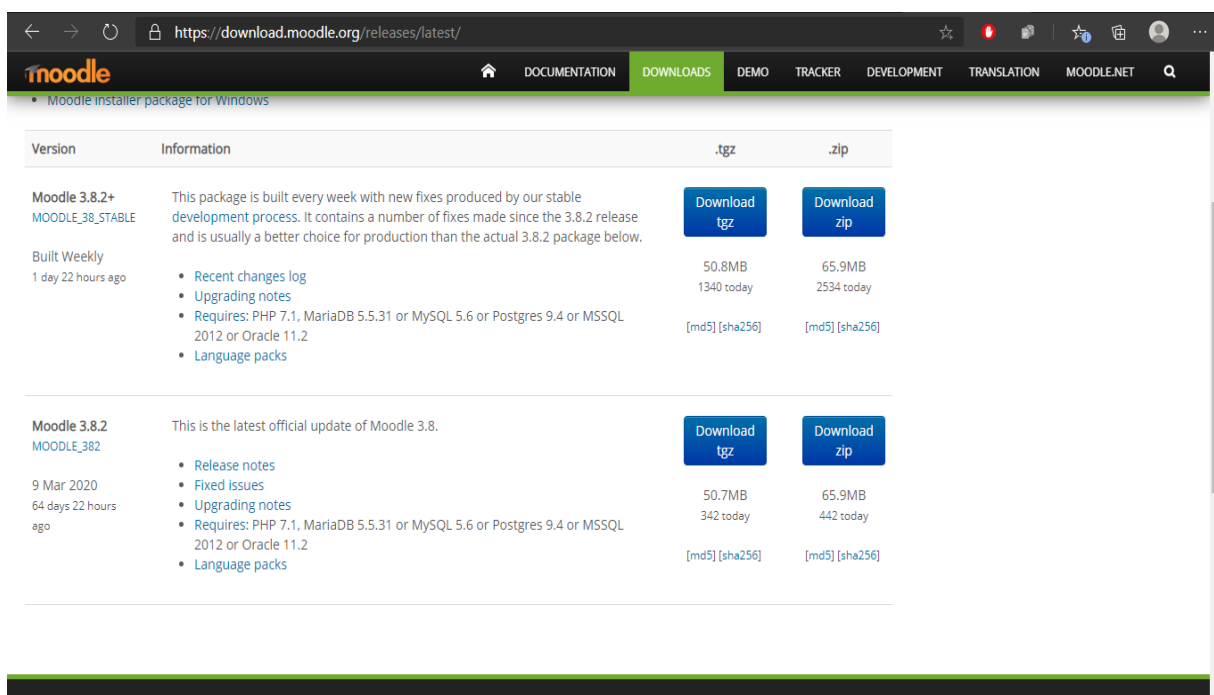


Рисунок 2.2 Сторінка завантаження архіву сервера

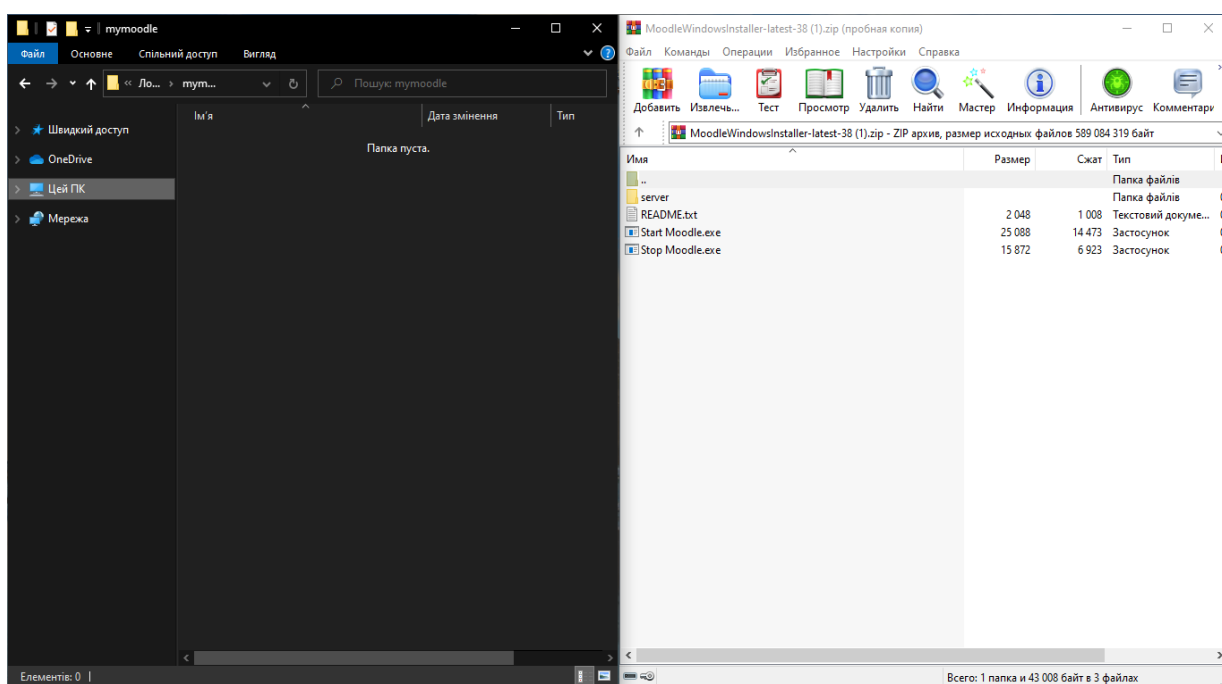


Рисунок 2.3 Витягнення всіх файлів архівів в потрібну директорію

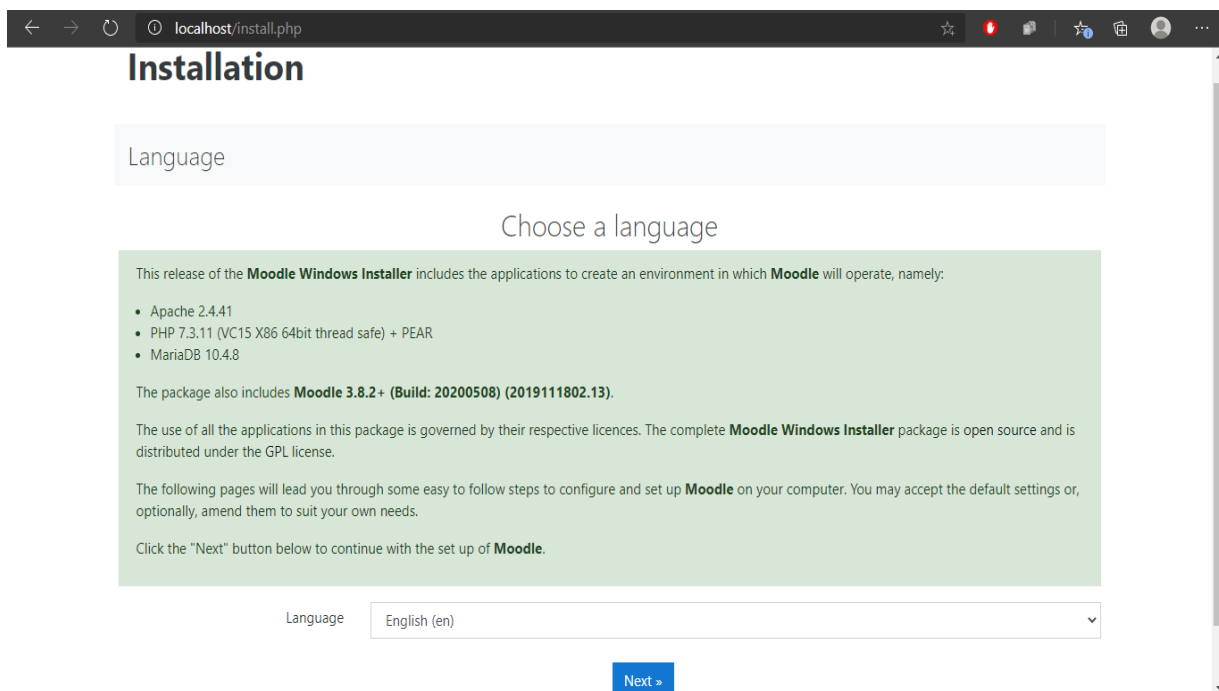


Рисунок 2.4 Перша сторінка встановлення серверу

Потім буде запропоновано обрати адресу за якою буде доступний наш сервер та директорії в якій повинні зберігатися його дані (див. рис. 2.5).

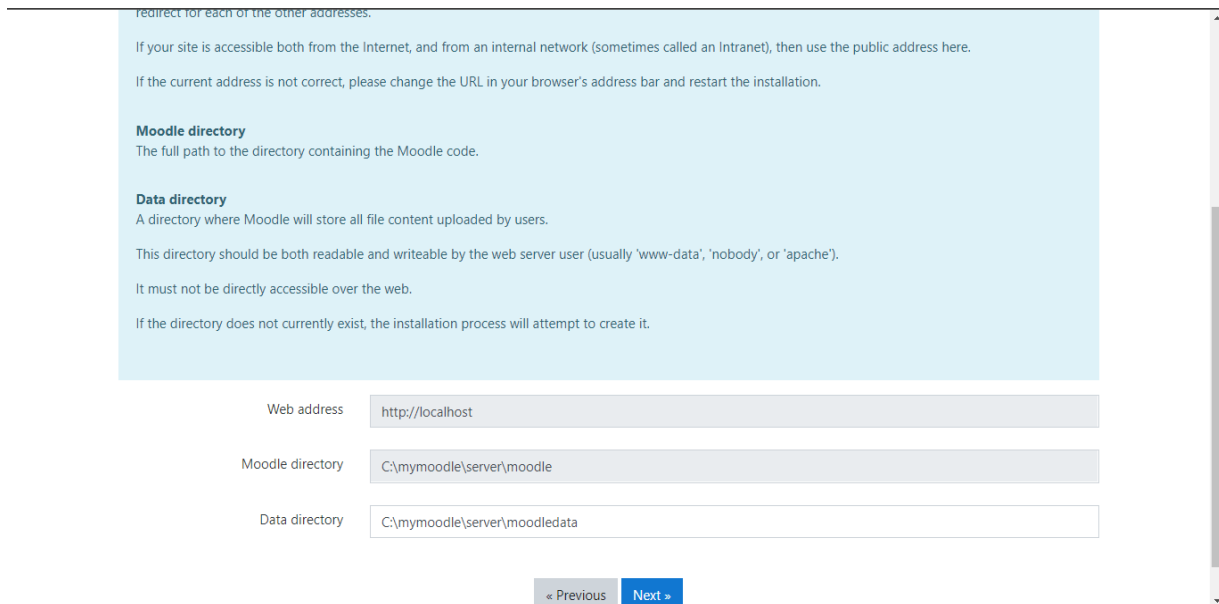


Рисунок 2.5 Сторінка вибору директорій

Далі відбувається налаштування бази даних сервера (див. рис. 2.6).

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		23

The database is where most of the Moodle settings and data are stored and must be configured here.

The database name, username, and password are required fields; table prefix is optional.

The database name may contain only alphanumeric characters, dollar (\$) and underscore (_).

If the database currently does not exist, and the user you specify has permission, Moodle will attempt to create a new database with the correct permissions and settings.

This driver is not compatible with legacy MyISAM engine.

Database host: localhost

Database name: moodle

Database user: root

Database password:

Tables prefix: mdl_

Database port:

« Previous Next »

Рисунок 2.6 Сторінка налаштування бази даних

Далі потрібно погодитися з користувацькою ліцензією (див. рис. 2.7).

Installation

Moodle - Modular Object-Oriented Dynamic Learning Environment

Copyright notice

Copyright (C) 1999 onwards Martin Dougiamas (<https://moodle.com>)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

See the Moodle License information page for full details: <https://docs.moodle.org/dev/License>

Confirm

Have you read these conditions and understood them?

Continue Cancel

Рисунок 2.7 Сторінка користувацької ліцензії

Наступним кроком відбудеться перевірка доступності та установка розширень та плагінів (див. рис. 2.8).

Installation - Moodle 3.8.2+ (Build: 20200508)

Moodle 3.8.2+ (Build: 20200508)

For information about this version of Moodle, please see the online Release Notes

Server checks

Name	Information	Report	Plugin	Status
unicode		must be installed and enabled		
database	mariadb (5.5.5-10.4.8-MariaDB)	version 5.5.31 is required and you are running 10.4.8		
php		version 7.1.0 is required and you are running 7.3.11		
pcreunicode		should be installed and enabled for best results		
php_extension	iconv	must be installed and enabled		
php_extension	mbstring	should be installed and enabled for best results		
php_extension	curl	must be installed and enabled		
php_extension	openssl	must be installed and enabled		
php_extension	tokenizer	should be installed and enabled for best results		
php_extension	xmlrpc	should be installed and enabled for best results		
php_extension	soap	should be installed and enabled for best results		
php_extension	ctype	must be installed and enabled		
php_extension	zip	must be installed and enabled		
php_extension	zlib	must be installed and enabled		

Рисунок 2.8 Сторінка встановлення компонентів серверу

На наступній сторінці ви повинні налаштувати свій основний обліковий запис адміністратора, який матиме повний контроль над сайтом. Пізніше ви можете створити більше облікових записів адміністратора (див. рис. 2.9)[12].

On this page you should configure your main administrator account which will have complete control over the site. Make sure you give it a secure username and password as well as a valid email address. You can create more admin accounts later on.


[Expand all](#)

▼ **General**

Username ?

Choose an authentication method ? Manual accounts

The password must have at least 8 characters, at least 1 digit(s), at least 1 lower case letter(s), at least 1 upper case letter(s), at least 1 non-alphanumeric character(s) such as *, -, or #

New password ! ? [Click to enter text](#) 

☐ Force password change ?

First name !

Surname !

Email address !

Email display ?

Рисунок 2.9 Сторінка створення облікового запису адміністратора









Потім відбувається налаштування сайту. Потрібно вказати його повне та скорочене ім'я, географічне розташування, налаштувати аутентифікацію та пошту для розсилки повідомлень (див. рис. 2.10).

New settings - Front page settings

Full site name fullname

Short name for site (eg single word) shortname

Front page summary summary

This summary can be displayed on the front page using the course/site summary block.

New settings - Location settings

Default timezone timezone Default: Australia/Perth

Рисунок 2.10 Сторінка налаштування сервера

Після виконання всіх цих дій ми отримаємо доступ до інструментів платформи Moodle. Тепер ми можемо створювати наш курс.

Якщо виникне потреба зупинити сервер, то для цього необхідно запустити файл “Stop Moodle.exe”, що знаходиться в тій самій директорії, що і “Start Moodle.exe”, яким ми запускали сервер[12].

2.3.2. Перенесення курсу на серверну платформу Moodle

Для перенесення курсу на платформу Moodle потрібно[12]:

- 1)Запустити Moodle сервер;
- 2)Зайти в свій обліковий запис;
- 3)У відповідному меню обрати пункт, що дозволяє створити новий курс;
- 4)Провести необхідне налаштування;
- 5)Перенести раніше підготовлений курс на платформу Moodle;
- 6)Привести отриманий результат до спільного формату та налаштувати загальний вигляд, дизайн.

Під час налаштування курсу можна вказати його назву, можливі дати початку та кінця курсу, його доступність, опис, формат і т.п. (див. рис. 2.11).

Рисунок 2.1 Сторінка створення курсу

Після початкових налаштувань стає доступним інтерфейс наповнення курсу матеріалом. Тут ви можете викладувати матеріали, створювати події, наприклад, обговорення чи тестування, додати словник чи обговорення в форматі форуму. Інструментів надзвичайно багато, як показано на рис. 2.12.

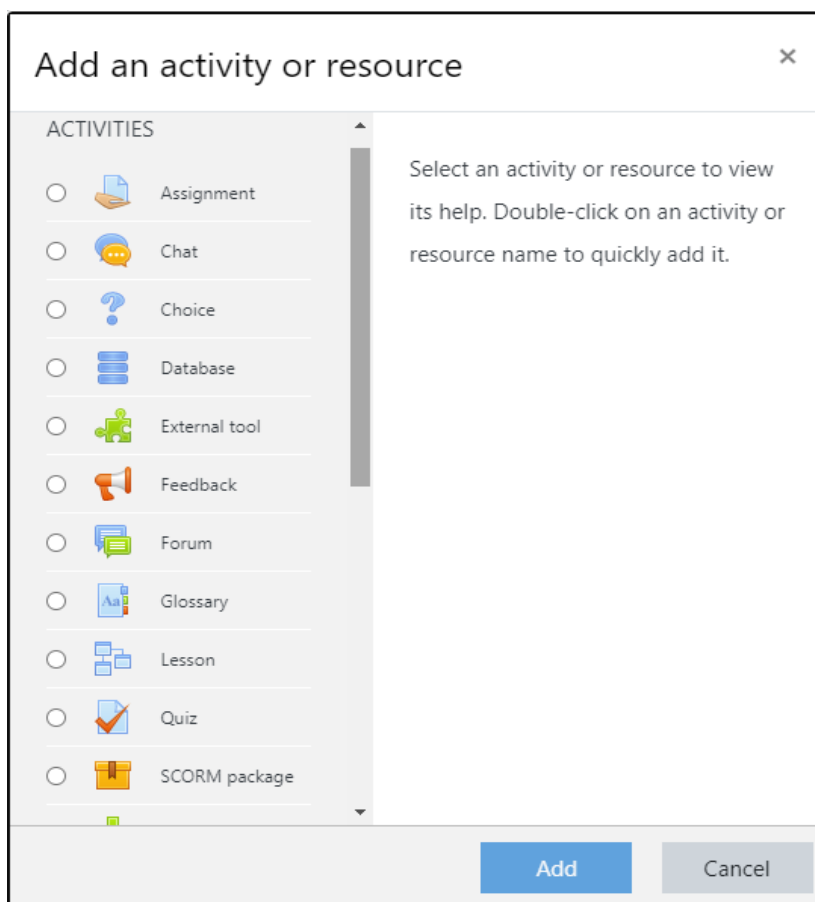


Рисунок 2.2 Меню додавання функціоналу в навчальний курс

ВИСНОВКИ ДО РОЗДІЛУ 2

В цьому розділі було зібрано та систематизовану всю необхідну інформацію для створення курсу, розроблена архітектура курсу, встановлено необхідне програмне забезпечення для реалізації курсу на обраній платформі.

					ДП 6430. 00.003 ПЗ	Арк
						30
Зм	Ар	№ докум.	Підпис	Дат		

РОЗДІЛ 3

РОЗРОБКА КУРСУ

Головною частиною розробки курсу є створення навчального матеріалу курсу. Відповідно до вище описаної архітектури (див. рис. 2.1), створення матеріалу курсу поділяється на створення вступної частини, основної та завершальної частин.

3.1. Розробка вступної частини

Основою вступної частини курсу є опис технологій, які стали основою для побудови SDN технології. Такими технологіями є: Amazon Web Services (AWS), Multi-tenant Data Center, Virtualized Multitenant Data Center, Virtual Local Area Network (VLANs), Ethernet Virtual Private Network (EVPN), Virtual Extensible LAN (VxLAN), Network Virtualization using Generic Routing Encapsulation (NVGRE), Multi-Protocol Label Switching (MPLS) та деякі інші[1][6]. Короткий опис всіх цих технологій міститься в розробленому курсі на Moodle. Як приклад, приведемо короткий опис технологій, що представлені в курсі:

Ідея веб-сервісів Amazon (AWS). Компанія зрозуміла, що може використовувати еластичні обчислювальні принципи для продажу невикористаного пулу ресурсів (сховища, обчислення та мережі), щоб його можна було використовувати з ефективністю, що наближається до 100%. Коли для внутрішніх користувачів потрібно більше ресурсів, вони просто відштовхують роздрібних користувачів, а коли їх немає, роздрібні комп'ютерні користувачі можуть використовувати вільні ресурси. Деякі називають це еластичними обчислювальними послугами або гіпервіртуалізацією через те, що більшість постійних клієнтів роблять це в досить великих масштабах і тому, що віртуалізація настільки поширена, що ця концепція використовується для зберігання, обчислення та зберігання ресурсів одночасно.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		31

Технологія VLANs. Коли виділяти мережеві ресурси потрібно лише для внутрішніх користувачів, і це число не перевищує 1000, VLANs є найпростішим та найефективнішим рішенням. Це добре відомий підхід, досить простий в експлуатації і підтримується на найрізноманітніших апаратних засобах, тому найпростішою формою мережевого накладання є, звичайно, плоска IP-мережа, яка проходить через підкладку VLAN. Насправді, так спочатку були побудовані DC (Data Center). В той час не було реальної потреби в підтримці кількох орендарів (користувачів), і коли з'явилася потреба у відокремленні ресурсів на основі відомчого доступу, були винайдені VLAN, але накладка все ще була відносно плоскою адресованою IP-мережею. Цей підхід навіть недовго працював для доступу зовнішніх користувачів до DC (як у випадку з Amazon AWS). Поки кількість орендарів не стала занадто великою, адресні простори повинні були перекриватися і загальна зміна цих віртуалізованих мережевих елементів вимагала дуже швидкого (пере)-програмування ресурсів. У таких сценаріях базовий IP через мережу Ethernet був достатнім для одного орендаря, і його можна було б легко та ненав'язливо розширити, щоб підтримувати до 4000 орендарів, що використовують VLAN (див. рис. 3.1).

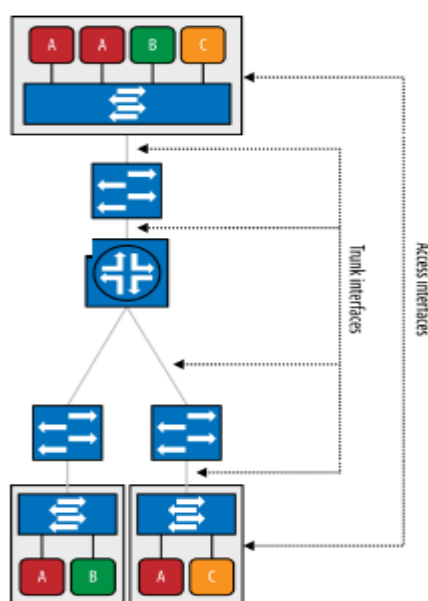


Рисунок 3.1 Принцип побудови VLAN

Рис. 3.1 ілюструє, як базовий підхід VLAN може використовуватися для реалізації локалізованого накладання центру обробки даних з накладенням IP та здійснення відносно простого з'єднання DC. У прикладі створено три 801.1q VLAN для кожної програми (додаток, база даних та управління), а трафік ділиться за допомогою цих VLAN по всій мережі. У цьому підході той самий плоский простір IP-адреси використовується в межах DC для того, щоб забезпечити доступ 3-го рівня до VM (virtual machine), що розміщують ці служби. Інтерфейс від віртуальної машини до гіпервізора - це інтерфейс доступу. Гіпервізор призначає інтерфейс VLAN орендаря. Інтерфейс комутатора сервер-доступ, а також усі інтерфейси перемикання на комутатори - це магістральні інтерфейси.

Хоча рішення VLAN досить легко імплементувати, якщо маршрутизація не вставлена між локальними мережами VLAN, давно відомо, що покладання просто на міст або протокол Spanning Tree Protocol (та його варіанти) не збільшує масштаби понад 300 хостів на будь-якій одній VLAN. Проблема з протоколом полягає в тому, що коли кількість MAC-адрес хостів стає великою, зміни, переміщення або збої призводять до масової обробки всередині мережевих елементів. Під час цих дуже напружених періодів мережеві елементи можуть пропускати інші збої або просто перевантажуватися до того моменту, коли вони не можуть налаштуватися досить швидко.

Це може призвести до мережевих циклів або чорних дір. Ранні версії Spanning Tree Protocol також страждали від проблеми витрачання рівних витрат (тобто паралельних) посилянь, оскільки протокол фактично блокував усі паралельні посилення на один міст, крім одного, як засіб запобігання циклів. Навіть останні версії цих протоколів мають обмежену здатність забезпечувати багатосторонні можливості. Нарешті, у більшості випадків мости можуть втрачати до декількох секунд при пересиланні, коли вони знову встановлюють зв'язок з іншими мостовими ланками.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		33

Технологія MPLS - Multi-Protocol Label Switching. “У 1980-х роках, коли Інтернет став популярним, дослідники почали досліджувати шляхи підвищення продуктивності систем пакетної обробки. Виникла ідея, яка в кінцевому підсумку була прийнята комерційними постачальниками: замінити підхід без комутації пакетів IP, який вимагає пошуку таблиць найдовших префіксів, на підхід, орієнтований на з'єднання, який забезпечує швидший алгоритм пошуку. Загальна концепція, відома як комутація міток, змусила виробників створити мережеву технологію, відому як Асинхронний режим передачі (Asynchronous Transfer Mode (ATM)). У 90-х роках ATM мав недовговічну популярність.

Причин можливого згасання популярності ATM було кілька. Основна причина була економічною: комутатори Ethernet та маршрутизатори IP були значно дешевшими, ніж комутатори ATM, і як тільки IETF створив технології комутації міток за допомогою звичайних IP-маршрутизаторів, IT-відділи не знайшли вагомих причин для ATM. Як тільки ми зрозуміємо загальні концепції, ми побачимо, як можна здійснити комутацію звичайним маршрутизатором, який не покладається на дороге обладнання, орієнтоване на з'єднання.

В основі перемикання лежить основне спостереження щодо пошуку: якщо в таблиці переадресації є N елементів, комп'ютер вимагає в середньому приблизно $\log_2 N$ кроків, щоб виконати збіг найдовшого префікса. Прихильники комутації міток зазначають, що апаратне забезпечення може виконувати індекс масиву в один крок. Крім того, індексація може переводитись безпосередньо в комбінаторне обладнання, тоді як пошук зазвичай включає кілька посилань на пам'ять.

Технології комутації використовують індексацію для досягнення надзвичайно швидкої переадресації. Для цього кожен пакет містить невелике ціле число, відоме як мітка. Коли пакет приходить до комутатора, комутатор витягує мітку і використовує значення як індекс у таблиці, яка вказує (точно визначає) відповідну дію. Кожен перемикач має набір

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		34

вихідних інтерфейсів, і дія зазвичай складається з відправлення пакету з одного з інтерфейсів.

Основний недолік описаної вище основної схеми комутації виникає через те, що кожна мітка складається з невеликого цілого числа. Як можна поширити схему на мережу, яка має багато напрямків? Ключовий варіант відповіді на питання включає переосмислення наших припущень щодо мереж та переадресації:

- Замість того, щоб орієнтуватися на пункти призначення, зосередьтеся на потоках пакетів.

- Замість того, щоб припускати, що таблиці переадресації залишаються статичними, уявіть собі систему, яка може швидко налаштувати або змінити таблиці переадресації.

Ми визначаємо потік пакетів як послідовність пакетів, що надсилаються з даного джерела до заданого пункту призначення. Наприклад, всі пакети на ТСП-з'єднанні складають єдиний потік. Потік також може відповідати телефонному дзвінку VoIP або довготривалій взаємодії. Ми можемо узагальнити ключову ідею:

Технології комутації використовують абстракцію потоку та створюють переадресацію потоків, а не пунктів призначення.

З точки зору потоків пакетів, ми припустимо, що існує механізм для створення записів в комутаторах, коли потік налаштований, і видалення записів, коли потік припиняється. На відміну від звичайного Інтернету, де таблиці переадресації залишаються статичними, очікується, що таблиці в комутаторах часто змінюються. Звичайно, система для налаштування потоків потребує механізмів, які розуміють місця призначення та шляхи їх досягнення. Таким чином, під час налаштування потоку може знадобитися розуміння звичайної переадресації, яка використовує призначення. Зараз ми зосередимось на роботі комутаторів і відкладемо обговорення налаштування потоку...”[1][6] – це лише короткий вступ в тему MPLS, в

курсі цій темі присвячено досить багато часу, а сама технологія описана досить детально, бо вона є значущою та цікавою.

Таким чином у вступній частині курсу описуються технологічні проблеми та концепції для вирішення котрих і була розроблена технологія SDN. Детальність опису кожної технології залежить від її значущості та складності, але їх об'єднує те, що вони всі описуються лише для знайомства з ними, а отже, матеріал курсу містить лише загальні принципи роботи і не вдається в незначні, для теми курсу, деталі цих технологій.

3.2. Розробка основної частини

На відміну від вступної частини курсу, де розглядалися технології, що не є ціллю курсу, в основній частині описується саме та інформація для якої і був створений цей курс, а отже, вона подається максимально повно, детально та зрозуміло. Для демонстрації можна привести приклад опису архітектури SDN контролера:

“Контролер SDN надає послуги, здатні реалізувати розподілену площину управління, а також керувати поняттями ефемерного стану (сприяє концепціям швидкозмінюваного стану) управління та централізації.

Насправді будь-який контролер забезпечить фрагмент або підмножину цієї функціональності. Ідеалізований контролер показаний на рис. 3.2.

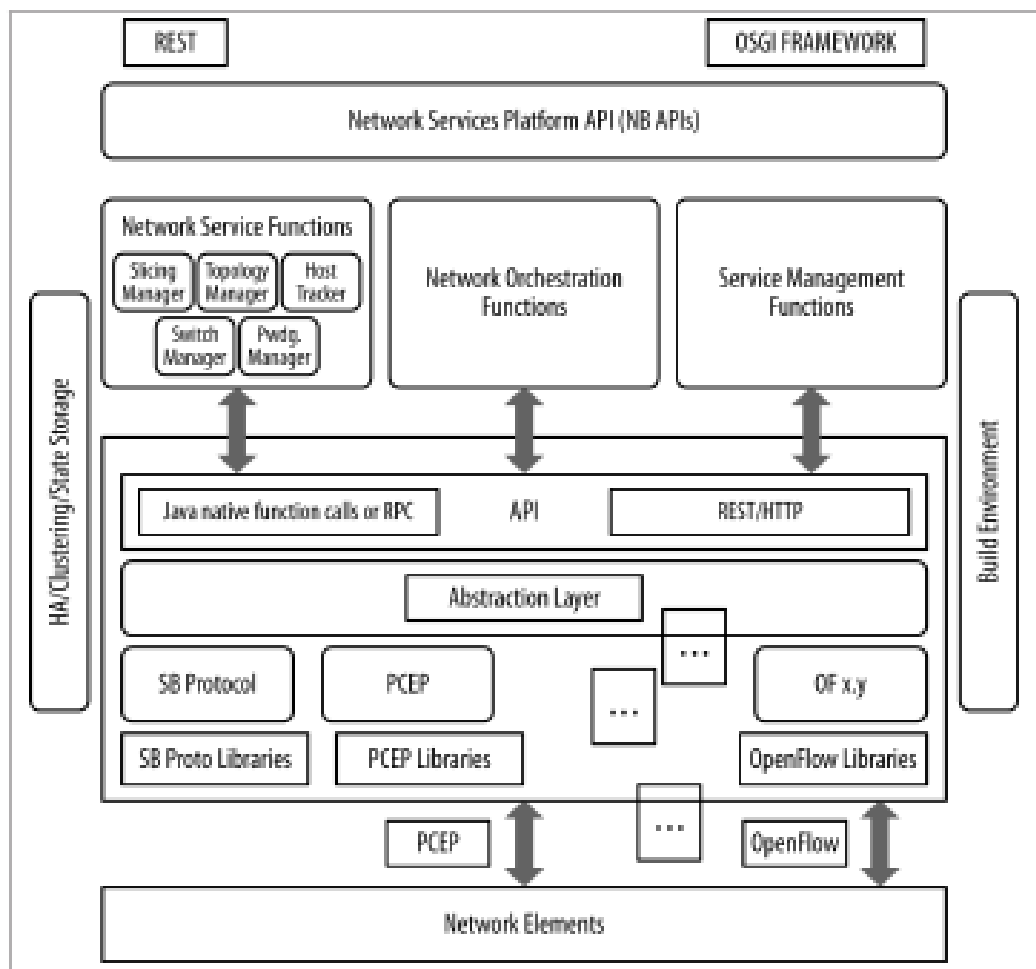


Рисунок 3.2 Ідеалізований SDN контролер

Архітектура SDN не визначає внутрішню розробку або реалізацію контролера SDN. Це може бути єдиний монолітний процес, або об'єднання ідентичних процесів, організованих для розподілу навантаження або захисту один одного від збоїв. В якості SDN контролер може виступати набір чітко функціональних компонентів у співпраці. Для деяких своїх функцій, наприклад, обчислення шляху, він може підписуватися на зовнішні сервіси. Дозволяється будь-яка комбінація цих альтернатив: контролер SDN розглядається як чорний ящик, визначений його зовнішньою поведінкою. Компоненти контролерів можуть виконуватися на довільних обчислювальних платформах. Вони також можуть виконуватись на розподілених та, можливо, мігруючих ресурсах, таких як віртуальні машини в центрах обробки даних.

Архітектура SDN контролера визначає зовнішню поведінку контролера, що відповідає принципам SDN.

Контролер SDN має деяку глобальну сферу застосування, його компоненти обмінюються інформацією стану, так що жоден зовнішній блок не повинен турбуватися про конфліктуючі чи суперечливі команди контролера. У міру того, як OSS впливає на ресурси або стани, вони мають однакові вимоги щодо координації з будь-якими контролерами SDN, які можуть бути залучені.

Кілька компонентів контролера можуть мати спільний доступ для запису до мережевих ресурсів, але, щоб відповідати принципам SDN, вони повинні:

а) бути налаштованим на управління конкретними наборами ресурсів або дій;

б) бути синхронізованими між собою, щоб вони ніколи не видавали непослідовні чи суперечливі команди.

Примітка 1 - При контролі розподілених обчислень розподілених мережевих ресурсів сувора синхронізація стану може спричинити за собою надмірні штрафи в продуктивності або складність.

Примітка 2 - Припущення про внутрішню узгодженість у контролері є окремим від питання про узгодженість стану з огляду на основні ресурси площини даних. Контролери SDN завжди очікують, що вони зможуть вирішувати пов'язані події, що асинхронно видимі з різних частин інфраструктури.

Не маючи наміру мінімізувати їх важливість, такі питання, як завантажувальна програма, синхронізація, міграція, резервне копіювання, аудит, управління випуском програмного забезпечення контролера тощо, є внутрішніми для контролера SDN в чорному ящику і не є предметом архітектури SDN.

Загальний опис SDN контролера - це програмна система або набір систем, що забезпечує:

- Управління станом мережі, а в деяких випадках управління та розповсюдження цього стану, може включати базу даних. Ці бази даних служать сховищем для інформації, отриманої з керованих мережевих елементів та відповідного програмного забезпечення, а також інформації, керованої програмами SDN, включаючи стан мережі, деяку ефемерну конфігураційну інформацію, вивчену топологію та інформацію про сеанси управління. У деяких випадках контролер може мати безліч цілеспрямованих процесів управління даними (наприклад, реляційні та нереляційні бази даних). В інших випадках можуть використовуватися й інші стратегії баз даних в пам'яті.

- Модель даних високого рівня, яка фіксує зв'язки між керованими ресурсами, політикою та іншими послугами, що надаються контролером. У багатьох випадках ці моделі даних побудовані за допомогою мови моделювання Yang.

- Забезпечення сучасного інтерфейсу програмування (API) програм (зазвичай RESTful - representational state transfer), який надає послуги контролера додаткам. Це полегшує більшість взаємодій між контролером та додатком. Цей інтерфейс ідеально відображається на моделі даних, яка описує послуги та функції контролера. У деяких випадках контролер та його API є частиною середовища розробки, що генерує код API з моделі. Деякі системи йдуть далі і забезпечують надійні середовища розробки, що дозволяють розширити основні можливості та подальше опублікування API для нових модулів, включаючи ті, що підтримують динамічне розширення можливостей контролера.

- Безпечний сеанс контролю TCP між контролером та пов'язаними агентами в мережевих елементах.

- Протокол на основі стандартів, що забезпечує стан мережі, керованої додатками, на мережевих елементах.

- Пристрій, топологія та механізм виявлення сервісів.

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		39

- Система обчислення шляхів та потенційно інші інформаційні послуги, орієнтовані на мережу або ресурси.

Нинішній набір контролерів включає комерційні продукти VMware (vCloud / vSphere), Nicira (NVP), NEC (Trema), Big Switch Networks (Floodlight / BNC) та Juniper / Contrail. Він також включає ряд контролерів з відкритим кодом.

Окрім використання OpenFlow та фірмових протоколів, існують контролери SDN, які використовують функціональні можливості мереж IP / MPLS для створення VPN-мереж MPLS 3 рівня, поділу орендарів для центру обробки даних або MPLS LSP для накладання в WAN.

Ми не можемо ігнорувати твердження про те, що контролери на базі NETCONF майже не можна відрізнити від рішень управління мережею або що контролери на основі Radius/Diameter, такі як PCRF та / або TDF, в мобільних середовищах також є контролерами SDN...”[3][4].

Для більш точного розуміння того, як саме відбувається подання інформації в основній частині курсу приведемо ще один приклад:

“Лічильники в OpenFlow Switch

Лічильники ведуться для кожної потокової таблиці, входу, порту, черги, групи, групового сегмента, лічильника та смуги лічильника. Лічильники, сумісні з OpenFlow, можуть бути реалізовані в програмному забезпеченні та підтримуватися за допомогою опитування апаратних лічильників з більш обмеженими діапазонами. Таблиця 1 містить набір лічильників, визначених специфікацією OpenFlow. Перемикач не потрібен для підтримки всіх лічильників, лише тих, що позначені "Required" в рис. 3.3.

Counter	Bits	
Per Flow Table		
Reference Count (active entries)	32	<i>Required</i>
Packet Lookups	64	<i>Optional</i>
Packet Matches	64	<i>Optional</i>
Per Flow Entry		
Received Packets	64	<i>Optional</i>
Received Bytes	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Port		
Received Packets	64	<i>Required</i>
Transmitted Packets	64	<i>Required</i>
Received Bytes	64	<i>Optional</i>
Transmitted Bytes	64	<i>Optional</i>
Receive Drops	64	<i>Optional</i>
Transmit Drops	64	<i>Optional</i>
Receive Errors	64	<i>Optional</i>
Transmit Errors	64	<i>Optional</i>
Receive Frame Alignment Errors	64	<i>Optional</i>
Receive Overrun Errors	64	<i>Optional</i>
Receive CRC Errors	64	<i>Optional</i>
Collisions	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Queue		
Transmit Packets	64	<i>Required</i>
Transmit Bytes	64	<i>Optional</i>
Transmit Overrun Errors	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Group		
Reference Count (flow entries)	32	<i>Optional</i>
Packet Count	64	<i>Optional</i>
Byte Count	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Group Bucket		
Packet Count	64	<i>Optional</i>
Byte Count	64	<i>Optional</i>
Per Meter		
Flow Count	32	<i>Optional</i>
Input Packet Count	64	<i>Optional</i>
Input Byte Count	64	<i>Optional</i>
Duration (seconds)	32	<i>Required</i>
Duration (nanoseconds)	32	<i>Optional</i>
Per Meter Band		
In Band Packet Count	64	<i>Optional</i>
In Band Byte Count	64	<i>Optional</i>

Рисунок 3.3 Список лічильників

Тривалість стосується часу, коли в комутаторі встановлено власний запис, порт, групу, чергу чи лічильник, який слід відслідковувати з секундною точністю. Поле помилок прийому - це загальна кількість всіх помилок прийому та зіткнень, визначених на рис. 3.3.

Пакетні лічильники для об'єкта OpenFlow, повинні рахувати кожен пакет, що використовує цей об'єкт, навіть якщо об'єкт не має впливу на пакет або якщо в кінцевому підсумку пакет відкидається або надсилається до контролера.

Наприклад, комутатор повинен підтримувати пов'язані з пакетом лічильники для наступного:

- власний запис лише з інструкцією goto-table та без дій;
- група, що виводить на неіснуючий порт;
- власний запис, що викликає виняток TTL;
- порт, який не працює.

Лічильники не підписані і працюють без додаткового індикатора. Лічильники повинні використати повний діапазон бітів, визначений для лічильника перед перекиданням, наприклад, якщо лічильник визначений як 64 біт, він не може використовувати лише нижні 32 біти. Якщо в комутаторі немає спеціального цифрового лічильника, його значення повинно бути встановлено в максимальне поле.

Групова таблиця

Групова таблиця складається з групових записів. Можливість потокового запису вказувати на групу дозволяє OpenFlow представляти додаткові методи переадресації (наприклад, select і all).

Group Identifier	Group Type	Counters	Action Buckets
------------------	------------	----------	----------------

Рисунок 3.4 Основні компоненти запису таблиці

Кожен запис групи (див. рис 3.4) ідентифікується ідентифікатором групи та містить:

- group identifier: 32-бітове ціле число без підпису, що однозначно ідентифікує групу на комутаторі OpenFlow;
- group type: для визначення групової семантики;
- counters (лічильники): оновлюються, коли пакети обробляються групою;
- action buckets: упорядкований перелік груп дій(action buckets, далі буде вказуватись, як відра), де кожне відро містить набір дій для виконання та

пов'язаних з ними параметрів. Дії у відрі завжди застосовуються як набір дій.

Груповий запис може складатися з нуля або більше відер, за винятком групи непрямого типу, яка завжди має одне відро. Відро, що не має дій, скидає пакет.

Відро зазвичай містить дії, які змінюють пакет, і вихідну дію, яка пересилає його до порту. Відро може також включати групову дію, яка викликає іншу групу, якщо комутатор підтримує групове поєднання, в цьому випадку обробка пакетів триває в групі, яку викликали. Відро без жодних дій є дійсним, відро без виводу або групової дії видаляє клон пакету, асоційованого з цим відром.

Типи груп

Перемикач не потрібен для підтримки всіх типів груп, лише тих, що зазначені нижче "Обов'язково". Контролер також може запитувати перемикач щодо того, який із типів "необов'язково" він підтримує.

- **Обов'язково: indirect (непрямо):** виконайте одне визначене відро у цій групі. Ця група підтримує лише одне відро. Дозволяє декільком власним записам або групам вказувати на загальний ідентифікатор групи, підтримуючи швидше і ефективніше конвергенцію (наприклад, наступні переходи для пересилання IP-адреси). Цей тип групи фактично ідентичний всій групі з одним відром. Ця група є найпростішим типом групи, і тому комутатори, як правило, підтримують більшу їх кількість, ніж інші типи груп.

- **Обов'язково: all (всі):** Виконати всі відра в групі. Ця група використовується для переадресації багатоадресної передачі або широкомовної передачі. Пакет ефективно клонується для кожного відра; один пакет обробляється для кожного окремого відра. Якщо відро направляє пакет явно через порт введення, цей клон пакету видаляється. Якщо записуючий контролер хоче переслати на порт входу, група повинна

включати додаткове відро, яка включає вихідну дію на зарезервованій порт OFPP_IN_PORT.

- Необов'язково: select (вибірний): Виконує одне відро в групі. Пакети обробляються одним відром у групі на основі алгоритму вибору, який обчислюється комутацією (наприклад, хеш на деякий налаштований користувачем кортеж або просто по колу). Всі налаштування та стан алгоритму вибору є зовнішніми для OpenFlow. Алгоритм вибору повинен забезпечувати рівний розподіл навантаження і, можливо, базуватися на вазі відра. Коли порт, визначений у відрі для групи вибору, відмовляє, комутатор може обмежувати вибір відра іншим набором (тим, у кого переадресуються дії, на вільні порти) замість того, щоб скидати пакети, призначені для цього порту. Така поведінка може зменшити відмови комутатора та його окремих частин.

- Необов'язково: fast failover (швидка відмова): виконати перше живе відро. Кожне відро пов'язане з певним портом та / або групою, яка контролює її життєдіяльність. Відра оцінюються в порядку, визначеному групою, і вибирається перше відро, пов'язане з живим портом / групою.

Цей тип групи дозволяє комутатору змінювати переадресацію, не вимагаючи зв'язку з контролером. Якщо немає відра, що задовольнить необхідні параметри, пакети скидаються. Цей тип групи повинен реалізувати механізм життєдіяльності.

Моніторинг життєдіяльності групи

Підтримка групи швидкої відмови вимагає моніторингу життєдіяльності для визначення конкретного відра для виконання.

Інші типи груп не вимагають здійснення моніторингу життєдіяльності, але можуть необов'язково здійснювати його. Якщо комутатор не може здійснити перевірку життєздатності для будь-якого відра в групі, він повинен відмовитись від модифікації групи та повернути помилку.

Групове відро може явно відстежувати життєдіяльність порту чи іншої групи. Правила визначення життєдіяльності включають:

					ДП 6430. 00.003 ПЗ	Арх
Зм	Ар	№ докум.	Підпис	Дат		44

- Порт вважається "живим", якщо він існує в шляху даних і має OFPPS_LIVE flag у своєму полі стану порту. Життєздатність порту може керуватися кодом за межами частини OpenFlow комутатора, визначеної поза специфікацією OpenFlow, наприклад, Spanning Tree або механізму KeepAlive. Порт не повинен вважатися активним (і OFPPS_LIVE flag повинен бути відключений), якщо один з механізмів життєдіяльності порту комутатора, включеного на цьому порту OpenFlow, вважає, що порт не працює, або якщо порт конфігує біт OFPPC_PORT_DOWN, означає, що порт відмовив, або якщо біт стану порту OFPPS_LINK_DOWN вказує, що посилення не працює.

- Відро вважається в прямому ефірі, якщо або watch_port не є OFPP_ANY, а переглянутий порт живе, або якщо watch_group не є OFPG_ANY, а група, яку переглядають, є активною. Іншими словами, відро вважається невідомим, якщо watch_port OFPP_ANY або переглянутий порт не живе, або якщо watch_group OFPG_ANY або група, яку переглянули, не працює.

- Група вважається живою, якщо хоча б одне її відро живе.

Контролер може зробити висновок про стан життєздатності групи, контролюючи стан різних портів.

Таблиця лічильників

Таблиця лічильників складається з записів лічильників, що визначають поточні лічильники. Поточні лічильники дозволяють OpenFlow впроваджувати обмеження швидкості, проста операція QoS, що обмежує набір потоків на обрану пропускну здатність.

Індивідуальні лічильники можуть також дозволити OpenFlow реалізовувати більш складні операції з контролем QoS, такі як DSCP (Differentiated Services Code Point - точка коду диференційованих послуг - елемент архітектури комп'ютерних мереж, що описує простий масштабований механізм класифікації, управління трафіком і забезпечення якості обслуговування. DSCP може застосовуватися, наприклад, для

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		45

зниження затримки чутливого мережевого трафіку на кшталт голосового зв'язку і потокового мультимедіа, тоді як інший трафік буде йти без пріоритизації. DSCP використовує 6-бітне поле 8-бітного IP-заголовка DS. Для IPv4 використовується простір застарілого заголовка ToS, який розділений на поля DSCP і ECN. У термінології IPv6 заголовок називається Traffic Class, логіка його організації абсолютно ідентична.) на основі вимірювання, що може класифікувати набір пакетів у декількох категоріях на основі їх швидкості. Лічильники повністю не залежать від черг на порт, однак у багатьох випадках ці дві функції можна поєднувати для реалізації складних робіт, що зберігають QoS frameworks, таких як DiffServ.

Лічильник вимірює швидкість призначених йому пакетів і дозволяє контролювати швидкість цих пакетів.

Лічильники приєднуються безпосередньо до поточкових записів (на відміну від черг, які прикріплені до портів). Якщо комутатор підтримує це, будь-який поточковий запис може вказати дію лічильника у списку дій: лічильник вимірює та контролює швидкість сукупності всіх поточкових записів, до яких він додається.



Рисунок 3.5 Основні компоненти запису лічильників в таблиці лічильників

Кожен запис лічильника (див. рис 3.5) ідентифікується його ідентифікатором лічильника і містить:

- meter identifier: 32-розрядне беззнакове ціле число, що ідентифікує лічильник;
- meter bands: невпорядкований список ліній лічильників, де кожна лінія лічильника вказує швидкість смуги і спосіб обробки пакету;
- counters: оновлюються, коли пакети обробляються лічильником(meter).

Різні записи в тій же таблиці можуть використовувати один і той же лічильник, різні лічильники або взагалі відсутні лічильники.

					ДП 6430. 00.003 ПЗ	Арх
Зм	Ар	№ докум.	Підпис	Дат		46

Використовуючи різні лічильники у потоковій таблиці, розмежований набір поточкових записів можна виміряти незалежно.

Пакети можуть проходити через кілька лічильників при використанні лічильників у послідовних таблицях, у кожній таблиці відповідний запис може спрямовувати його до одного лічильника. Ще один спосіб використання декількох лічильників, якщо комутатор підтримує його, є використання декількох дій лічильника за кожний запис. Це можна використовувати для здійснення ієрархічного вимірювання, коли різні набори трафіку вперше вимірюються незалежно, а потім разом (див. Рис.3.6).

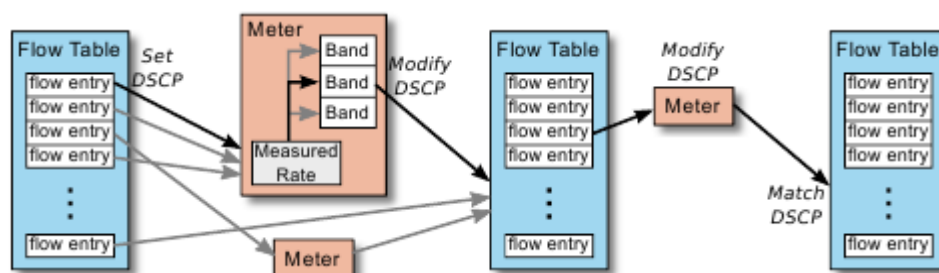


Рисунок 3.6 Вимірювачі та ієрархічне вимірювання DSCP

З цих прикладів можна зробити висновок, що в основній частині курсу інформації подається набагато більше, вона є більш деталізованою, та надається більше пояснень. Всі ці тези можна не одразу помітити в наведених прикладах тексту курсу, бо це лише вирізані частини з досить об'ємних тем та тому, що вся інформація надається в максимально простому та зрозумілому форматі.

3.3. Розробка завершальної частини

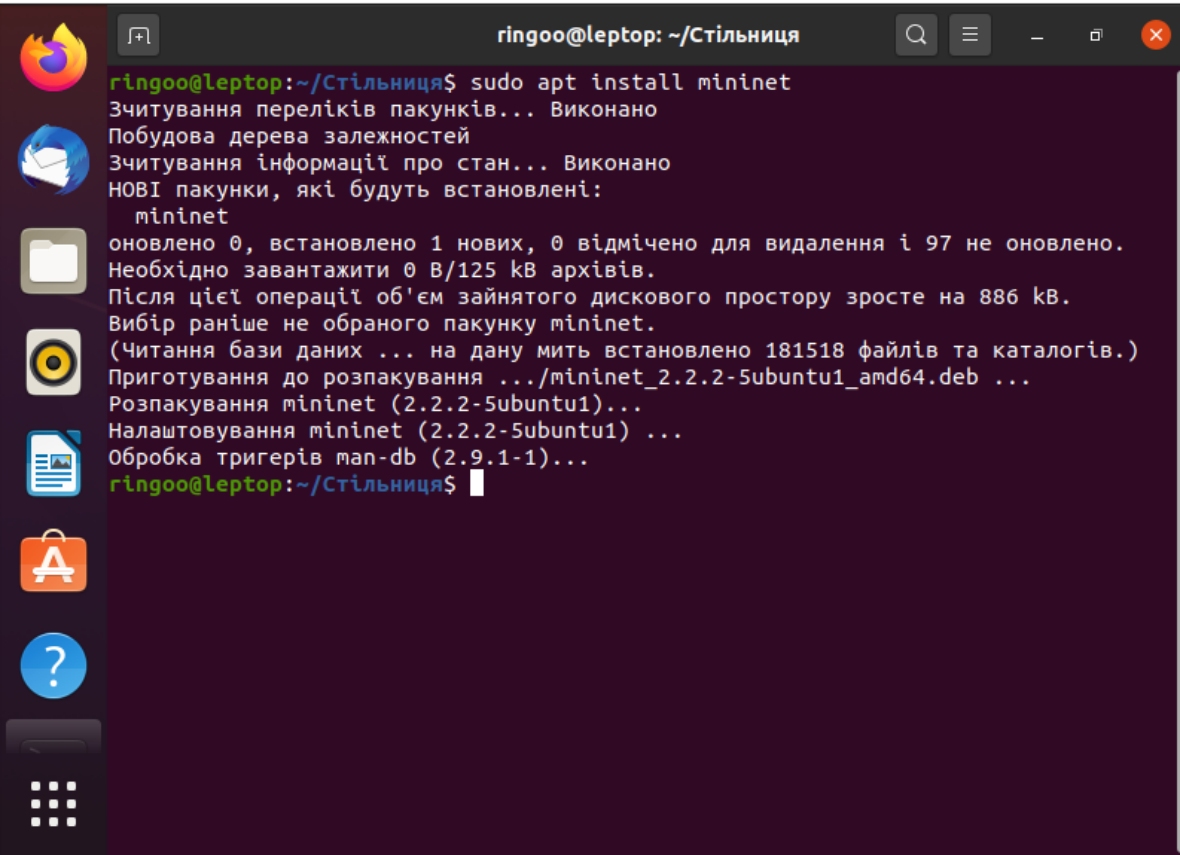
В завершальній частині курсу принцип подачі інформації залишається таким самим як і в основній частині, змінюється сама інформація. Тут акцентується увага на використанні технології SDN, а не на її принципі роботи.

В якості прикладу приведемо вступ в ту частину, де в відбувається знайомство з програмою Mininet, що дозволяє створити реалістичну віртуальну мережу.

“Mininet - це емулятор мережі, що створює мережу віртуальних хостів, контролерів, комутаторів та каналів зв'язку. Mininet хости запускають стандартне мережеве програмне забезпечення Linux, його комутатори підтримують OpenFlow для дуже гнучкої маршрутизації та використання SDN[13].

Mininet дозволяє проводити навчання, дослідження, прототипування, розробку, тестування та будь-які інші завдання для яких потрібна повна експериментальна мережа.

Для встановлення Mininet необхідно в терміналі виконати команду “sudo apt install mininet”



```
ringoo@leptop: ~/Стільниця
ringoo@leptop:~/Стільниця$ sudo apt install mininet
Зчитування переліків пакунків... Виконано
Побудова дерева залежностей
Зчитування інформації про стан... Виконано
НОВІ пакунки, які будуть встановлені:
  mininet
оновлено 0, встановлено 1 нових, 0 відмічено для видалення і 97 не оновлено.
Необхідно завантажити 0 B/125 kB архівів.
Після цієї операції об'єм зайнятого дискового простору зросте на 886 kB.
Вибір раніше не обраного пакету mininet.
(Читання бази даних ... на дану мить встановлено 181518 файлів та каталогів.)
Приготування до розпакування .../mininet_2.2.2-Subuntu1_amd64.deb ...
Розпакування mininet (2.2.2-Subuntu1)...
Налаштовування mininet (2.2.2-Subuntu1) ...
Обробка тригерів man-db (2.9.1-1)...
ringoo@leptop:~/Стільниця$
```

Рисунок 3.3.1 Встановлення програми Mininet

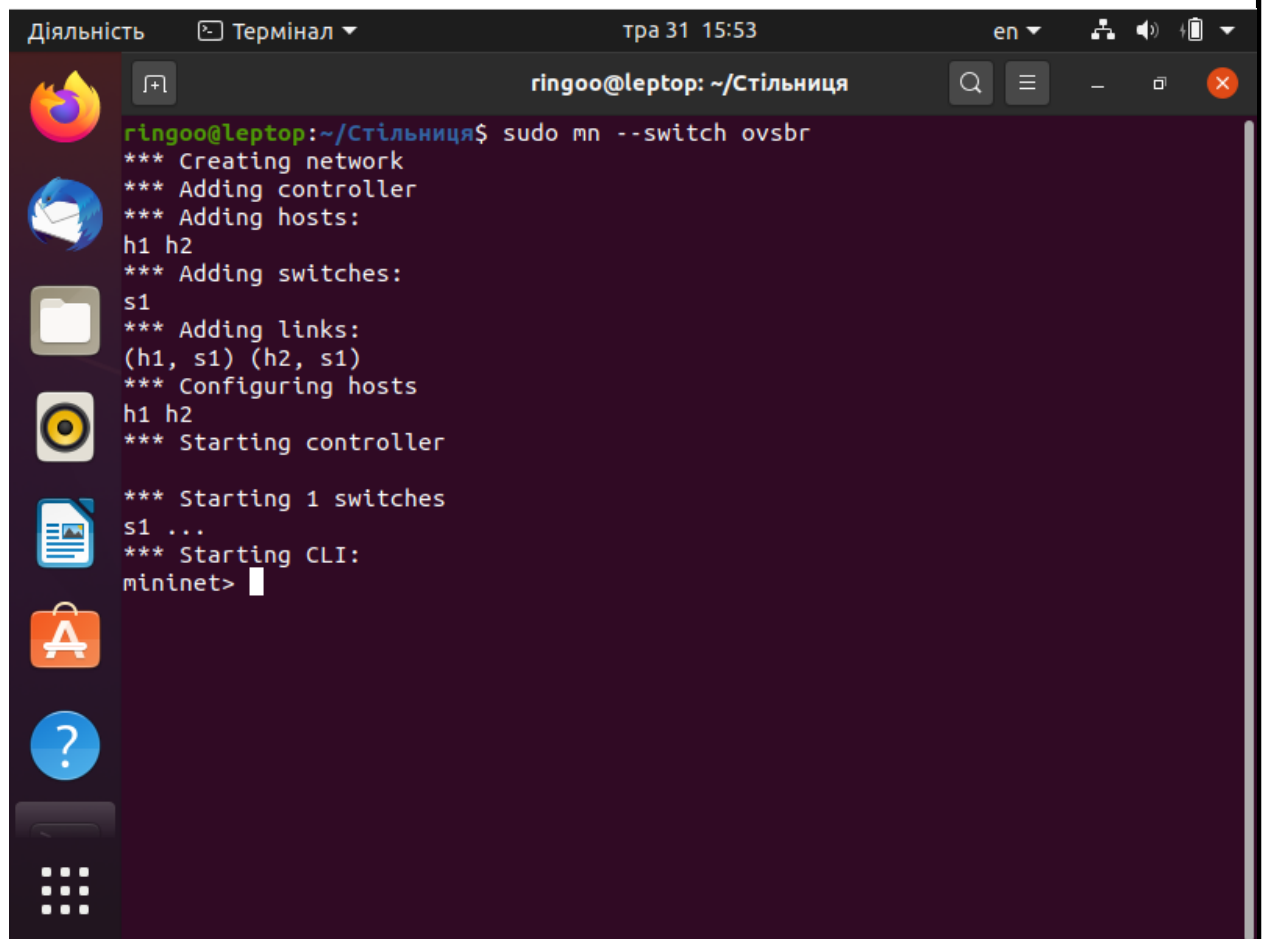
Після встановлення можна починати працювати з програмою.

Для вивчення всіх можливостей програмного забезпечення можна відвідати сторінку покрокової інструкції “<http://mininet.org/walkthrough/>” з офіційного сайту.”[13]

Також в якості прикладу можна привести частину курсу в якій відбувається знайомство користувача з основними командами програмного забезпечення[13]:

“Основні команди

Інструменти програмного забезпечення Mininet досить потужні. В якості знайомства ми розглянемо лише декілька основних команд. Розпочати варто з команди “mn”, яка створює готову мінімальну топологію мережі.



```
ringoo@leptop: ~/Стільниця
ringoo@leptop:~/Стільниця$ sudo mn --switch ovsbr
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Рисунок 3.3.2 Використання команди “mn”

На рис. 3.3.2 ця команда продемонстрована з параметром “--switch”, що встановлює базовий перемикач.

Наступними командами розглянемо “nodes”, “net” та “ifconfig”.

Nodes – відображає вузли топології.

Net – відображає зв’язки між вузлами топології.

Ifconfig – відображає інформацію про адреси вузла.

Також до основних можна віднести команди: “up”/“down”, які, відповідно, підключають/відключають порт мережевого інтерфейсу, “route”, що відображає таблиці маршрутизації, “ping”, що проводить перевірку зв’язку між визначеними вузлами та “pingall”, що перевіряє зв’язок між всіма вузлами, “iperf”, що проводить перевірку пропускної здатності мережі.

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo: s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> nodes
available nodes are:
h1 h2 s1
mininet> h2 ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.2 netmask 255.0.0.0 broadcast 10.255.255.255
    inet6 fe80::7815:1ff:fe7b:d902 prefixlen 64 scopeid 0x20<link>
    ether 7a:15:01:7b:d9:02 txqueuelen 1000 (Ethernet)
    RX packets 36 bytes 3980 (3.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 866 (866.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Локальна петля (Loopback))
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 3.3.3 Команди "net", "nodes", "ifconfig"

Список всіх команд та пояснення того, як вони роблять можна знайти на офіційному сайті “<http://mininet.org/>”.

З приведених прикладів можна зробити висновок, що в завершальній частині курсу інформація подається досить детально, але максимально

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		50

зрозуміло, а сама інформація стосується більше використання технології SDN, а не її архітектури, принципів роботи і т.п.

					ДП 6430. 00.003 ПЗ	Арк
						51
Зм	Ар	№ докум.	Підпис	Дат		

ВИСНОВКИ ДО РОЗДІЛУ 3

З використанням всіх доступних інструментів, підготовлені матеріали курсу для обраної платформи. На доповнення до тексту використовувалася велика кількість графічних зображень для більш наглядного пояснення матеріалу. До кожного розділу додано словник з основними термінами та обговорення в форматі форуму, що дозволить налаштувати зв'язок з особами, що будуть проходити курс, та дозволить відповідати на їх запитання. Також форум дозволить створити живе обговорення, що може призвести до поглиблення знань та розуміння теми усіх учасників обговорення.

					ДП 6430. 00.003 ПЗ	Арк
						52
Зм	Ар	№ докум.	Підпис	Дат		

РОЗДІЛ 4

ІНСТРУКЦІЯ З ЕКСПЛУАТАЦІЇ

4.1 Доступ до курсу

Для доступу до курсу вам необхідно мати доступ до інтернету та браузер, що дасть вам змогу зайти на сайт курсу та пройти процедуру входу в акаунт як це показано на рис. 4.1[12].

sdncourse

Username

Password

☐ Remember username

Log in

Forgotten your username or password?

Cookies must be enabled in your browser ?

Some courses may allow guest access

Log in as a guest

Is this your first time here?

For full access to this site, you first need to create an account.

Create new account


Рисунок 4.1 Сторінка входу

Для входу необхідно заповнити поля логіна та пароля та натиснути кнопку входу. Якщо ви хочете, щоб наступного разу ван не довелося заповняти ці поля то натисніть на поле, яке увімкне функцію запам'ятовування користувача.


					ДП 6430. 00.003 ПЗ	Арх
Зм	Ар	№ докум.	Підпис	Дат		53

Якщо акаунта у користувача нема, то його потрібно створити. Для цього потрібно заповнити поля у формі реєстрації.


▼ Choose your username and password

Username 


The password must have at least 8 characters, at least 1 digit(s), at least 1 lower case letter(s), at least 1 upper case letter(s), at least 1 non-alphanumeric character(s) such as as *, -, or #


Password 


▼ More details

Email 

address

Email (again) 

First name 

Surname 

City/town

Country

Select a country

⌵

Create my new account

Cancel

Рисунок 4.2 Сторінка реєстрації

Для реєстрації потрібно вказати свої:

- Обов'язково:
- Логін
- Пароль

- Електронну пошту (два поля)
- Ім'я
- Прізвище
- Опціонально:
- Місто
- Країна

4.2 Інструкція користувача

Пройшовши процедуру входу ви перейдете на сторінку де будуть показані всі доступні курси (див. рис. 4.3)[12].

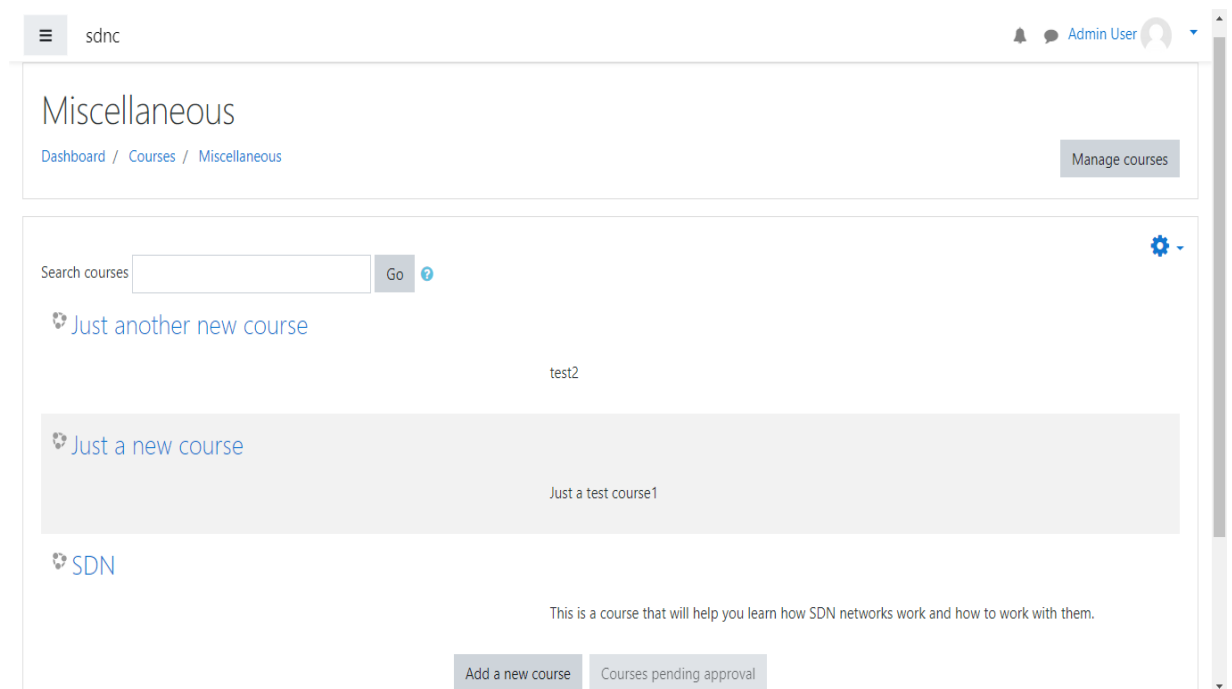


Рисунок 4.3 Сторінка вибору курсу

Серед цього списку потрібно обрати необхідний вам курс. Після цього ви попадете на сторінку курсу. Там ви, в залежності від обраного курсу, зможете побачити список тем, форум, чат і т. п. У випадку мого курсу ви побачите: теми, матеріали, форум та словник в кожній з тем.

Важливо розуміти, що хоч курс і може використовуватися як єдине джерело інформації з даної теми, все ж було б набагато ефективніше, з

					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		55

точки зору отримання знань та розуміння теми, також використовувати інші матеріали, в їх числі і ті, що вказані в курсі.

4.3 Недоліки та удосконалення

До недоліків даного курсу можна віднести загальну проблему будь-яких навчальних матеріалів: не для всіх осіб інформація, подана в даному курсі, буде однаково зрозумілою. Удосконалити цей аспект можна лише шляхом внесення правок відповідно до отримуваних відгуків та запитань користувачів. Також до проблем можна віднести те, що у сучасному світі технології настільки швидко змінюються та розвиваються, що будь-який навчальний матеріал швидко втрачає свою актуальність, але на відміну від книг чи статей, що випускаються у паперовому варіанті, курс, що реалізовано в електронному варіанті, може бути модифіковано, оновлено та актуалізовано.

					ДП 6430. 00.003 ПЗ	Арк
						56
Зм	Ар	№ докум.	Підпис	Дат		

ВИСНОВКИ ДО РОЗДІЛУ 4

В даному розділі було описано шлях, що дозволяє використати даний проект для отримання необхідних знань та навичок для створення та використання SDN мереж. Будь-який навчальний матеріал швидко втрачає свою актуальність, але на відміну від книг чи статей, що випускаються у паперовому варіанті, курс, що реалізовано в електронному варіанті, може бути модифіковано, оновлено та актуалізовано.

					ДП 6430. 00.003 ПЗ	Арк
						57
Зм	Ар	№ докум.	Підпис	Дат		

ВИСНОВКИ

Темою даного дипломного проекту є розробка програмних засобів навчання системних адміністраторів використанню можливостей SDN мереж. Даний проект створений для полегшення вивчення можливостей SDN мереж.

Були розглянуті книги, статті та інші навчальні матеріали і визначено сильні і слабкі сторони існуючих навчальних засобів вивчення SDN мереж. Було визначено, як можна зробити рішення навчального засобу, що вбере в себе всі сильні сторони існуючих засобів та буде позбавлене їх недоліків.

В якості платформи для створення навчального курсу було обрано Moodle. Moodle це одна з найпопулярніших платформ для навчання, що розповсюджується на відкритій основі.

Результатом роботи став навчальний курс, що дозволяє особі, що його пройде зрозуміти логіку розвитку мереж та передумови виникнення технології SDN мереж, їх сильні та слабкі сторони, принципи роботи та особливості створення і обслуговування таких мереж.

					ДП 6430. 00.003 ПЗ	Арк
						58
Зм	Ар	№ докум.	Підпис	Дат		

ПЕРЕЛІК ПОСИЛАНЬ

1. Руслан Смелянский. Программно-конфигурируемые сети. Открытые системы. СУБД, №9 (07.11.2012).
2. Kate Greene. 10 Emerging Technologies of 2009. Software Defined Networking. MIT Technology Review. MIT (24.02.2009).
3. Doug Marschke. Software Defined Networking (SDN): Anatomy of OpenFlow Volume I, 178 сторінок, Lulu Publishing Services (25.03.2015)
4. Thomas D. Nadeau, SDN: Software Defined Networks, 384 сторінки, O'Reilly Media (07.09.2013 p.)
5. SDN controller (software-defined networking controller) [З мережі]
<https://searchnetworking.techtarget.com/definition/SDN-controller-software-defined-networking-controller>
6. А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник, Комп'ютерні мережі, 256 сторінок, Магнолія 2006 (2013 рік)
7. SDN in Windows Server overview [З мережі] <https://docs.microsoft.com/en-us/windows-server/networking/sdn/>
8. Open Networking Foundation, Framework for SDN: Scope and Requirements [З мережі] https://www.opennetworking.org/wp-content/uploads/2014/10/Framework_for_SDN_Scope_and_Requirements.pdf
9. Software-defined networking [З мережі]
https://en.wikipedia.org/wiki/Software-defined_networking
10. Fei Hu, Network Innovation through OpenFlow and SDN: Principles and Design, 520 сторінок, CRC Press (18.02.2014)
11. Sangam Racherla, David Cain, Scott Irwin, Per Ljungström, Pushkar Patil, Alessio M. Tarenzio, Implementing IBM Software Defined Network for Virtual Environments, 284 сторінки, IBM Redbooks (04.09.2014 p.)
12. Курсы по работе с СДО Moodle [З мережі] <https://moodle.ru/>
13. Mininet Walkthrough [З мережі] <http://mininet.org/walkthrough/>

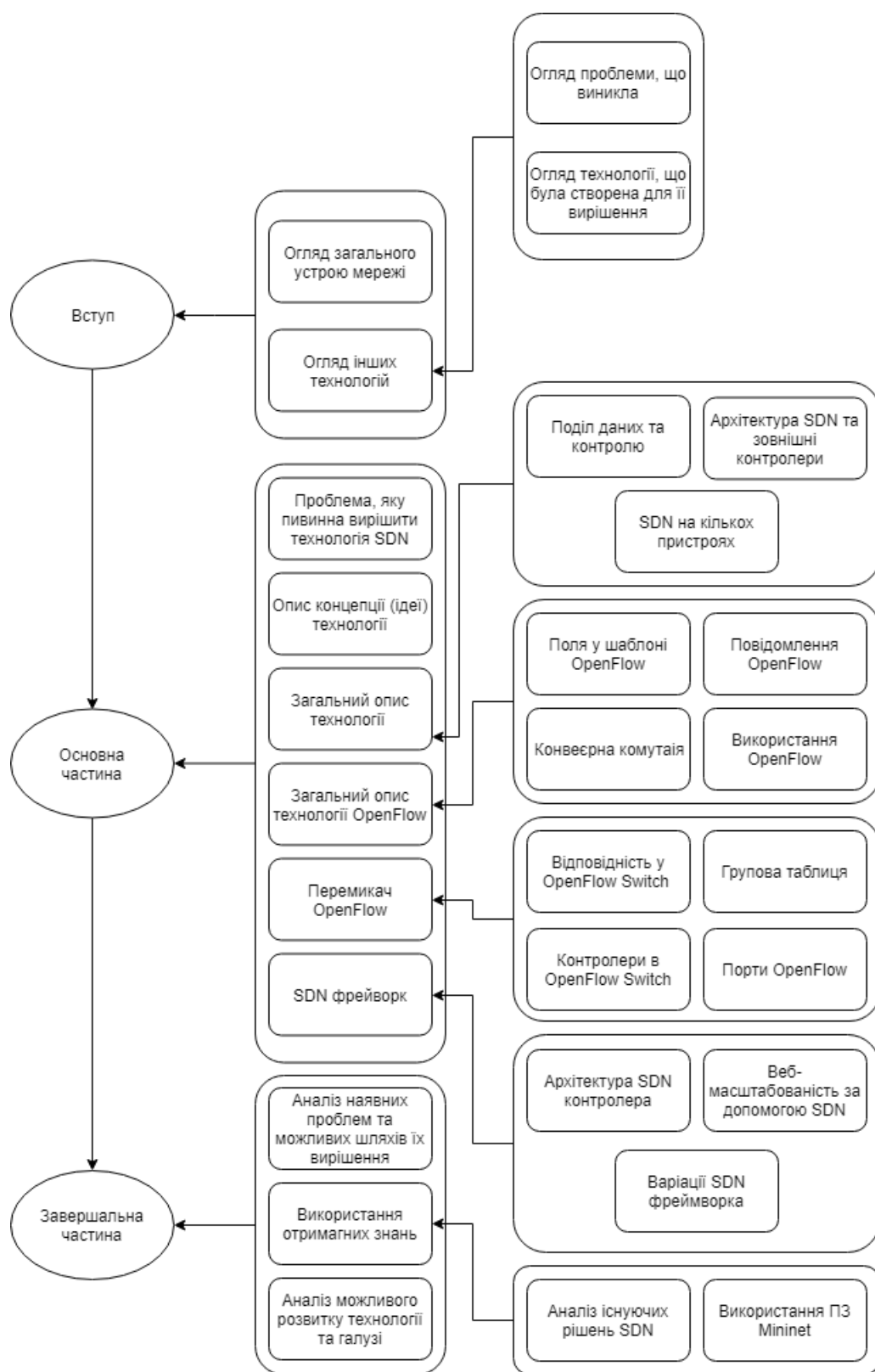
					ДП 6430. 00.003 ПЗ	Арк
Зм	Ар	№ докум.	Підпис	Дат		59

ДОДАТОК 1

«Програмні засоби навчання системних адміністраторів
використанню можливостей технологій SDN мереж»

СТРУКТУРА КУРСУ АРКУШІВ 1

Київ – 2020



ДП 6430. 00.004 Д1

Зм.	Арк	№ докум.	Підпис	Дата
		Шапран А.С.		
Перевірив.		Долголенко О.М.		
Н. Контр.		Симоненко В.П.		

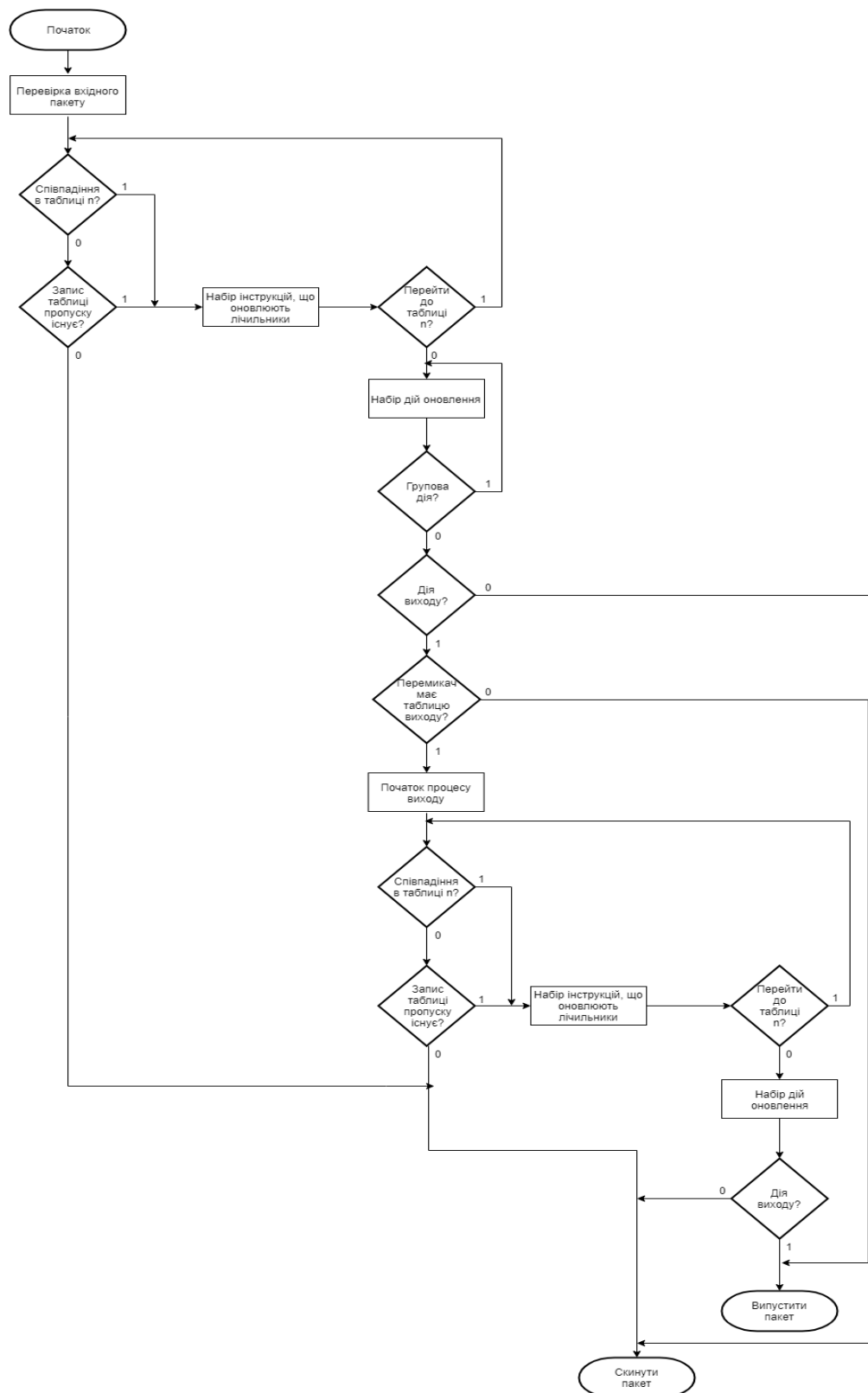
Структура курсу.

Літ.	Аркуш	Аркушів
	1	1
НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64		

ДОДАТОК 2

«Програмні засоби навчання системних адміністраторів
використанню можливостей технологій SDN мереж»

СПРОЩЕНИЙ БЛОК КЛАСИФІКАЦІЇ ПАКЕТІВ ЧЕРЕЗ КОМУТАТОР OPENFLOW АРКУШІВ 1



ДП 6430. 00.005 Д2

Зм.	Арк	№ докум.	Підпис	Дата
		Шапран А.С.		
Перевірив.		Долголенко О.М.		
Н. Контр.		Симоненко В.П.		

Спрощений блок
класифікації
пакетів через
комутатор
OpenFlow.

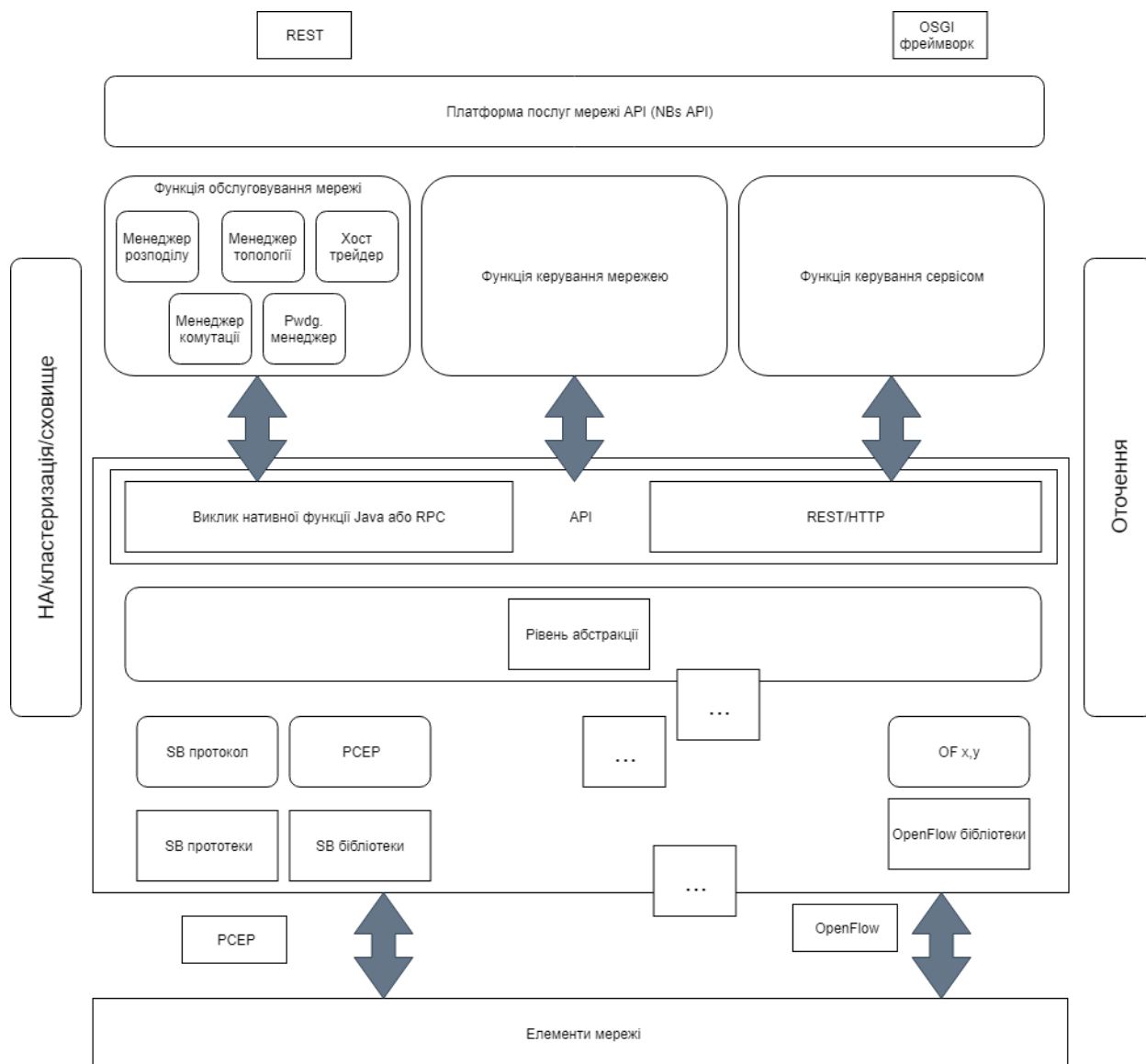
Літ.	Аркуш	Аркушів
	1	1
НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ зр. ІІІ-64		

ДОДАТОК 3

«Програмні засоби навчання системних адміністраторів
використанню можливостей технологій SDN мереж»

ІДЕАЛІЗОВАНИЙ SDN КОНТРОЛЕР АРКУШІВ 1

Київ – 2020



					ДП 6430. 00.006 ДЗ			
Зм.	Арк	№ докум.	Підпис	Дата				
		Шапран А.С.			Ідеалізований SDN контролер.			
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
						Літ.	Аркуш	Аркушів
							1	1
						НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64		

ДОДАТОК 4

«Програмні засоби навчання системних адміністраторів
використанню можливостей технологій SDN мереж»

ЧАСТИНА НАВЧАЛЬНОГО КУРСУ ПРИСВЯЧЕНА
ВСТУПУ В ТЕМУ SDN
АРКУШІВ 10

Концептуально мережевий пристрій, такий як маршрутизатор або комутатор, можна розділити на дві частини: механізми, що дозволяють менеджерам налаштовувати та керувати пристроєм та механізмами, що управляють пакетами. Для фіксації дихотомії (протипоставлення) ми використовуємо терміни площина управління та площина даних. Площина даних виконує всю обробку та пересилання пакетів. Площина управління забезпечує інтерфейс управління. Рисунок Д4.1 ілюструє концептуальний поділ.

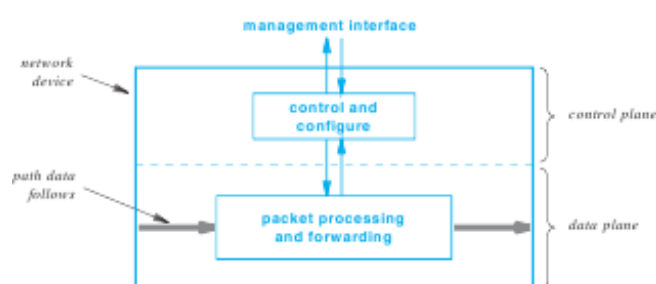


Рисунок Д4.1 Ілюстрація двох концептуальних частин мережевого пристрою: площини управління та даних.

Як показано на рисунку, з'єднання, за якими пакети надходять та відходять, мають значно більшу ємність, ніж з'єднання управління. Ми використовуємо термін шлях даних для позначення шляху великої ємності для пакетного трафіку та шляху управління для позначення шляху з меншою ємністю, який менеджер використовує для управління пристроєм.

На жаль, коли постачальники створюють мережеві пристрої, вони зазвичай дотримуються високо інтегрованого підходу, в якому площина управління пристрою та площина даних щільно з'єднані. Пристрій експортує інтерфейс управління, який дозволяє менеджеру контролювати

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата				
		Шапран А.С.			Частина навчального курсу присвячена темі вступу в SDN.			
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
					Літ. Аркуш Аркушів 1 1 НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64			

певні функції. Наприклад, інтерфейс постачальника на комутаторі VLAN дозволяє менеджеру визначати набір VLAN та асоціювати даний порт з однією з VLAN.

Аналогічно, інтерфейс постачальника на маршрутизаторі дозволяє менеджеру заповнювати записи таблиці переадресації. Однак менеджер не може налаштувати конкретні правила класифікації або контролювати обробку окремих пакетів.

Для досягнення гібридного рішення ми повинні знайти спосіб розділити площину даних і площину управління. Тобто нам потрібен спосіб замінити систему контрольної площини постачальника на індивідуальну (спеціалізовану) версію. Наша нова система управління повинна мати прямий доступ до обладнання апаратної площини даних і повинна бути в змозі налаштувати обробку та переадресацію пакетів. Рисунок Д4.2 ілюструє концепцію: нова мережа управління додана до мережевого пристрою:

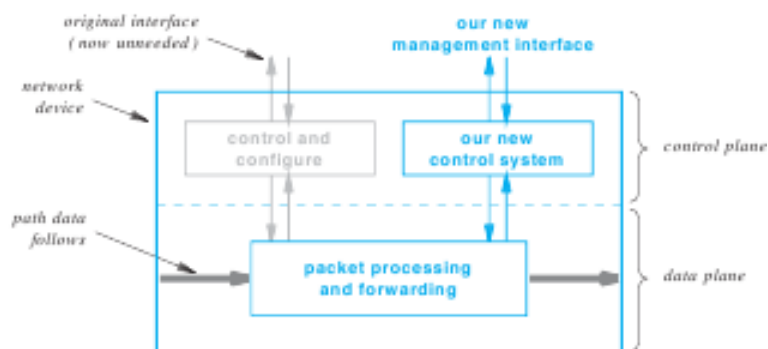


Рисунок Д4.2 Концептуальна організація пристрою з новою системою управління, що замінює систему постачальника.

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата	<p>Частина</p> <p>навчального курсу</p> <p>присвячена темі</p> <p>вступу в SDN.</p>			
		Шапран А.С.						
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
					<p>Літ.</p> <p>Аркуш</p> <p>Аркушів</p>			
							1	1
					<p>НТУУ "КПІ ім.</p> <p>Ігоря Сікорського"</p> <p>ФІОТ гр. ІІІ-64</p>			

Теоретично можливо встановити нову систему управління в мережевий пристрій без додавання додаткового обладнання. Щоб зрозуміти чому, зауважте, що більшість систем управління реалізовані в програмному забезпеченні: вбудований процесор виконує керуючу програму з ПЗУ. Таким чином, систему управління можна змінити, завантаживши в ПЗУ нове програмне забезпечення. На практиці замінити програмне забезпечення управління постачальника неможливо з двох причин.

По-перше, технологія SDN спирається на звичайну конфігурацію та переадресацію. По-друге, оскільки багато завдань, що виконуються програмним забезпеченням управління, характерні для базового обладнання, площина управління повинна бути спеціалізованою для кожного апаратного пристрою. Справа в тому, що хоча SDN-технологія потребує нової функціональної площини управління в мережевих пристроях, повністю замінити програмне забезпечення управління постачальника недоцільно.

Архітектура SDN та зовнішні контролери

Підхід, прийнятий для SDN, відокремлює програмне забезпечення управління від базових мережевих пристроїв. Замість того, щоб повністю переписати програмне забезпечення постачальника для управління, SDN використовує розширення підхід: програмне забезпечення SDN працює у зовнішній системі, а до пристрою додається невеликий модуль, який дозволяє зовнішній системі SDN налаштувати базове обладнання. Зовнішня система, як правило, звичайний ПК, називається контролером. Рисунок Д.А.3 ілюструє архітектуру SDN.

Як показано на малюнку, додавання зовнішнього контролера розширює функціональність площини управління поза пристроєм. Новий модуль SDN

					<i>ДП 6430. 00.007 Д4</i>		
<i>Зм.</i>	<i>Арк</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>			
		<i>Шапран А.С.</i>			<i>Частина навчального курсу присвячена темі вступу в SDN.</i>		
<i>Перевірив.</i>		<i>Долголенко О.М.</i>					
<i>Н. Контр.</i>		<i>Симоненко В.П.</i>					
						<i>Літ.</i>	<i>Аркуш</i>
							<i>Аркушів</i>
							1
							1
						<i>НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64</i>	

повинен бути доданий до площини управління пристрою, щоб дозволити зовнішньому контролеру налаштувати площину даних. На малюнку зображено модуль SDN як невеликий, оскільки код не містить розвинутої логіки, а також не забезпечує звичайний інтерфейс управління. Натомість модуль SDN просто приймає команди низького рівня від зовнішнього контролера і передає кожну команду до блоку обробки площини даних. Справа в тому, що модуль SDN мінімалістичний - він містить значно меншу складність та функціональність, ніж типовий механізм управління площиною.

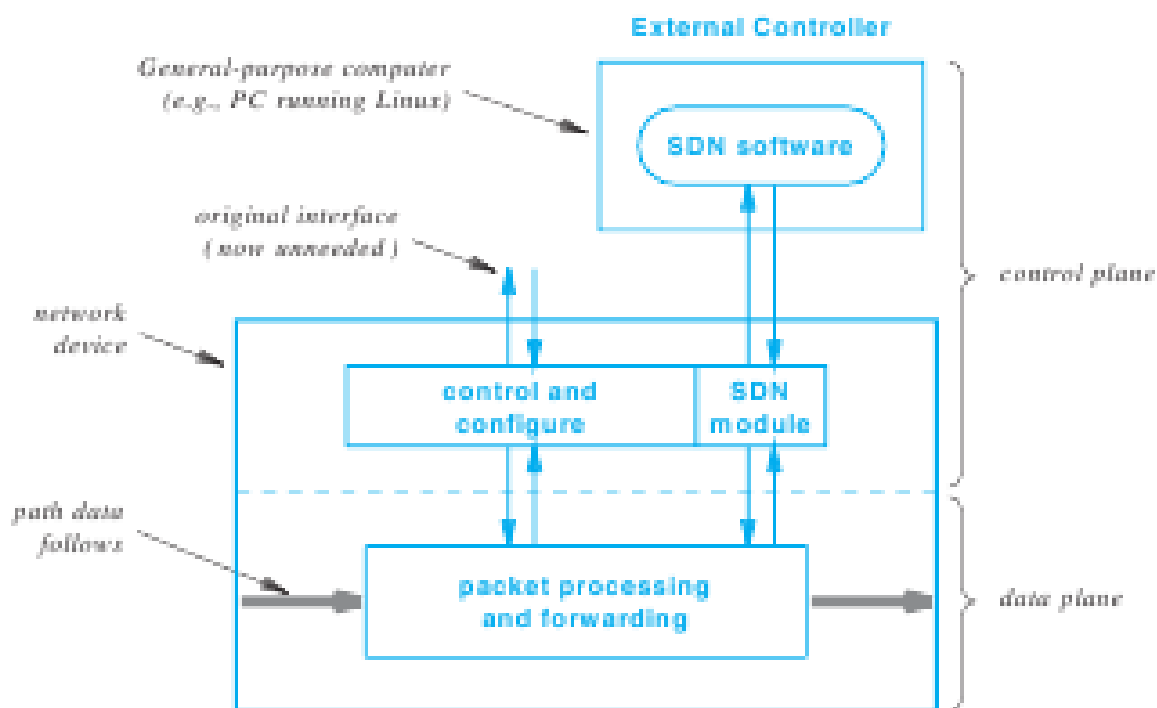


Рисунок Д4.3 Основна архітектура SDN з зовнішнім контролером, який налаштовує обладнання для класифікації та переадресації в мережевому пристрої.

					ДП 6430. 00.007 Д4		
Зм.	Арк	№ докум.	Підпис	Дата			
		Шапран А.С.			Частина навчального курсу присвячена темі вступу в SDN.		
Перевірив.		Долголенко О.М.					
Н. Контр.		Симоненко В.П.					
					Літ.		
					Аркуш		
					Аркушів		
					1		
					1		
					НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64		

Переміщуючи складні функції площини управління на зовнішній контролер, підхід SDN надає менеджерам більше контролю. Менеджер може налаштувати будь-які правила класифікації та переадресації, які дозволяє програмне забезпечення SDN, навіть якщо вони відрізняються від звичайних правил VLAN або IP підмережі, дозволених програмним забезпеченням постачальника. Ми побачимо, що менеджер може вибрати, як класифікувати та пересилати кожен окремий пакет. По суті, SDN використовує обладнання для класифікації та переадресації мережевого пристрою, але ігнорує площину управління, яку надає постачальник. Ідею можна узагальнити: підхід SDN відокремлює обробку площини управління від обробки площини даних шляхом переміщення функцій управління у зовнішній контролер. Оскільки він видає команди низького рівня, які налаштовують площину даних пристрою, зовнішній контролер може запропонувати менеджеру більше контролю, ніж програмне забезпечення площини управління постачальника.

SDN на кількох пристроях

В описі вище описано основний механізм, що пояснює, як технологію SDN можна використовувати для налаштування та управління одним мережевим пристроєм. Для забезпечення значної пропускну здатності в Інтернеті технологія повинна бути розширена на декілька мережевих пристроїв (наприклад, на всіх пристроях в кампусі або в мережі Інтернет-провайдера). Важлива ідея полягає в тому, що ми хочемо, щоб всі контролери були підключені до мережі, що дозволить керувати прикладним програмним забезпеченням, яке працює на контролерах, спілкуватися та координуватись на всьому наборі пристроїв.

					ДП 6430. 00.007 Д4		
Зм.	Арк	№ докум.	Підпис	Дата	Частина навчального курсу присвячена темі вступу в SDN.		
		Шапран А.С.					
Перевірів.		Долголенко О.М.					
Н. Контр.		Симоненко В.П.					
					Літ. Аркуш Аркушів 1 1 		
					НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64		

Два основні питання стосуються загальної архітектури: скільки контролерів потрібно, і як вони повинні бути пов'язані між собою? Як можна уявити, кількість необхідних зовнішніх контролерів залежить від типу мережевих пристроїв, якими керують, та програмного забезпечення SDN. Якщо мережеві пристрої невеликі (наприклад, модеми), а комп'ютер, який використовується як контролер, є потужним, один контролер може працювати з багатьма пристроями.

Однак якщо певному пристрою потрібна система управління для обробки багатьох винятків або внесення частих змін у конфігурацію, контролер може працювати лише з одним мережевим пристроєм. Зараз ми будемо вважати, що кожен мережевий пристрій має виділений контролер; пізніше ми розглянемо розширення, де контролер може працювати з декількома пристроями.

Щодо спілкування між контролерами, ми уявимо, що існує окрема мережа управління для підключення набору контролерів. Контролери використовують мережу управління для спілкування один з одним, а це означає, що ми можемо створити програмне забезпечення для управління, яке працює на контролерах і координує весь набір пристроїв. Щоб дозволити менеджеру управління встановлювати політику та виконувати інші завдання управління, ми вважатимемо, що мережа управління включає також комп'ютер менеджера. Рисунок Д4.4 ілюструє ідеалізовану версію мережі управління, яка з'єднує між собою контролери:

					ДП 6430. 00.007 Д4						
Зм.	Арк	№ докум.	Підпис	Дата	Частина навчального курсу присвячена темі вступу в SDN.						
		Шапран А.С.									
Перевірів.		Долголенко О.М.									
Н. Контр.		Симоненко В.П.									
					Літ.			Аркуш		Аркушів	
								1		1	
					НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64						

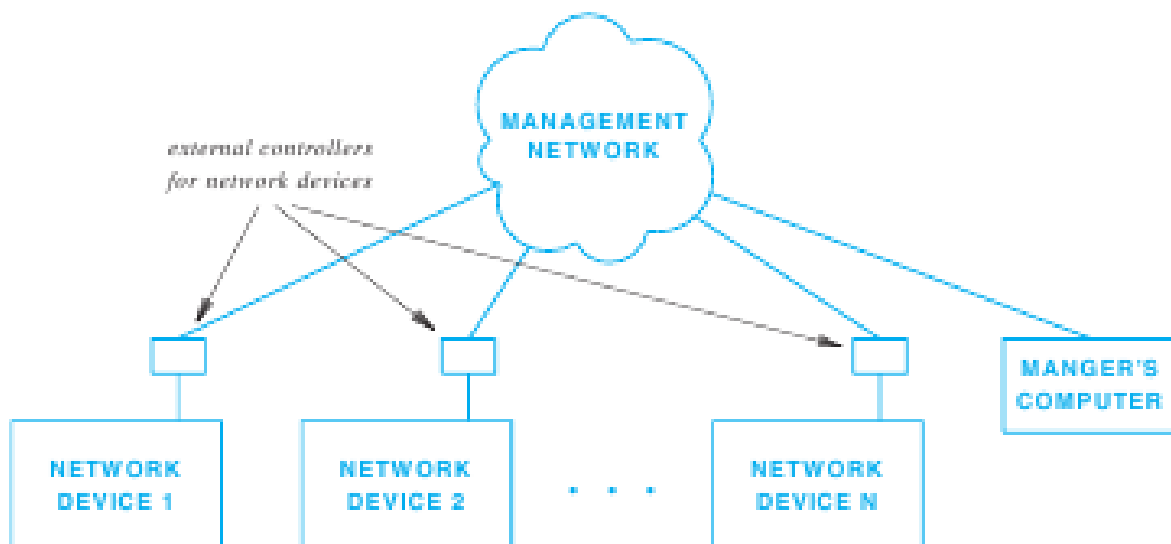


Рисунок Д4.4 Ідеалізований взаємозв'язок контролерів SDN в окрему мережу управління. Програми управління на контролерах використовують мережу управління для координації між собою.

Оскільки він відображає лише мережу управління, цей рисунок оминає важливий фрагмент архітектури: мережі передачі даних, якими керує. Не бачачи жодної з мереж передачі даних, може бути важко зрозуміти, чому потрібно підключити контролери. Відповідь полягає в тому, що мережеві пристрої, якими керують, діляться мережами передачі даних. Наприклад, припустимо, що пристрій 1 і пристрій 2 обидва є комутаторами VLAN, які мають пряме з'єднання. Далі припустимо, що два комутатори розміщені спільно у великій університетській лабораторії, де вони з'єднують набір комп'ютерів. Якщо менеджер налаштовує VLAN для лабораторії, VLAN повинен охоплювати обидва комутатори, а конфігурації повинні бути узгоджені.

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата				
		Шапран А.С.			Частина навчального курсу присвячена темі вступу в SDN.			
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
					Літ. Аркуш Аркушів			
					1 1			
					НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64			

Коли використовується підхід SDN, програми управління, що працюють у двох контролерах, повинні координуватися; для цього програми керування спілкуються через мережу управління.

Впровадження SDN за допомогою звичайних комутаторів

Мабуть, найцікавіший аспект технології SDN виникає внаслідок інтеграції системи управління та мережі передачі даних. SDN використовує той самий підхід, що і SNMP: трафік управління проходить через ті ж дроти, що і трафік даних. Тобто, замість того, щоб використовувати фізично окрему мережу, система управління використовує мережу, якою керує.

Щоб зрозуміти парадигму SDN, розгляньте фізичне з'єднання між комутатором Ethernet та контролером SDN для комутатора. Замість використання спеціалізованого апаратного інтерфейсу контролер може підключитися до стандартного порту Ethernet на комутаторі.

Звичайно, комутатор повинен бути налаштований на визнання контролера як привілейованого, щоб запобігти прийманню команд від довільного комп'ютера. Рисунок Д4.5 ілюструє найпростіше можливе розташування: прямий зв'язок між контролером і комутатором.

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата	Частина навчального курсу присвячена темі вступу в SDN.			
		Шапран А.С.						
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
					Літ. Аркуш Аркушів 1 1 НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64			

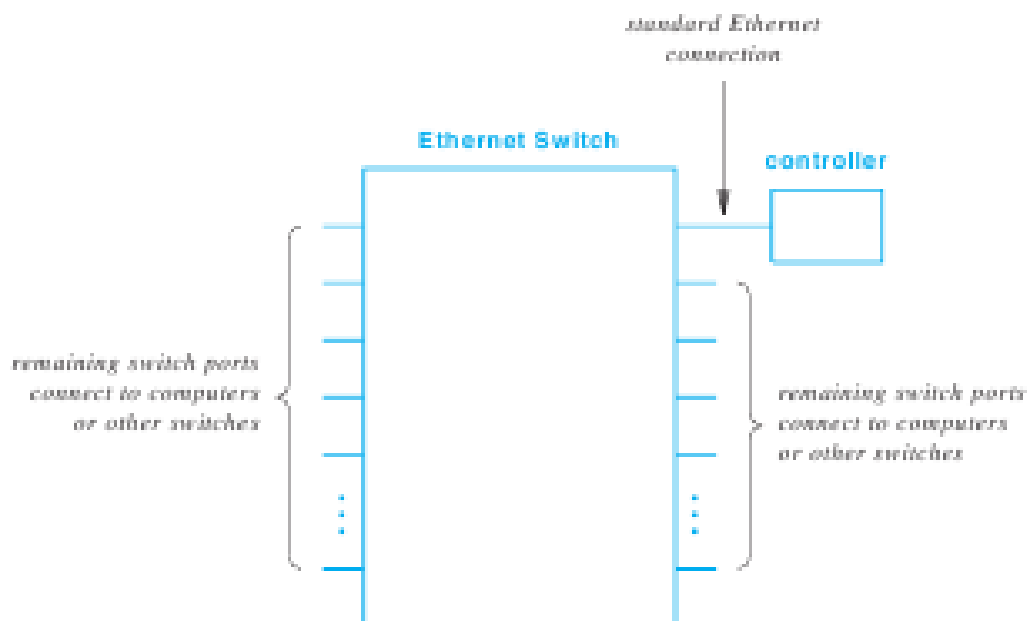


Рисунок Д4.5 З'єднання між перемикачем Ethernet та контролером SDN для комутатора.

Ідею можна узагальнити. Зауважте, що більшість внутрішніх мереж містять кілька комутаторів. Якщо уявити звичайну внутрішню мережу, переадресація рівня 2 та рівня 3 в комутаторах влаштована таким чином, що дозволяє контролеру, підключеному до комутатора, спілкуватися з іншими комутаторами. Наприклад, кожному комутатору буде призначена IP-адреса, і переадресація буде налаштована так, що комп'ютер може надсилати IP-дейтаграму на будь-який комутатор.

SDN припускає, що така конфігурація була встановлена до того, як програмне забезпечення SDN взяло під контроль мережу. Таким чином, контролер може використовувати IP для доступу до будь-якого мережевого пристрою. По суті, мережа управління на малюнку 4 є віртуальною накладкою на звичайну мережу, а не окремою фізичною мережею. Як і у

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата				
		Шапран А.С.			Частина навчального курсу присвячена темі вступу в SDN.		Літ.	Аркуш
Перевірив.		Долголенко О.М.						1
Н. Контр.		Симоненко В.П.						
						НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64		

випадку з SNMP, менеджери та програми управління, що використовують SDN, повинні бути обережними, щоб зберегти з'єднання накладки управління - неправильні зміни правил переадресації можуть залишати контролер не в змозі спілкуватися з одним або декількома комутаторами.

Технологія OpenFlow

Виникає кілька питань. Які саме можливості конфігурації та керування повинні запропонувати комутатор на зовнішній контролер SDN? Під час надсилання запиту SDN на комутатор, який формат повинен використовувати контролер? Як комутатор може розрізняти запити SDN, призначені для самого комутатора, та інші пакети, які передає контролер (наприклад, пакети, що надсилаються іншим контролерам)? Коротше кажучи, який протокол слід використовувати між контролером та комутатором?

Відповідь на питання лежить у протоколі, відомому як OpenFlow. Спочатку створений в Стенфордському університеті як спосіб дослідникам експериментувати з новими мережевими протоколами, OpenFlow отримав широке визнання. Багато постачальників комутаторів погодилися додати можливості OpenFlow до своїх комутаторів.

Оскільки на площині управління в більшості комутаторів працює операційна система, модуль OpenFlow можна легко додати до комутатора. Модуль OpenFlow працює точно так само, як модуль SDN, показаний на малюнку 3, більша частина інтелекту розміщена у зовнішньому контролері, а модуль у комутаторі виступає лише посередником, який переводить повідомлення від зовнішнього контролера в команди, передані в обладнання апаратної площини.

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата				
		Шапран А.С.			Частина навчального курсу присвячена темі вступу в SDN.			
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
						Літ.	Аркуш	Аркушів
							1	1
						НТУУ "КПІ ім. Ігоря Сікорського" ФІОТ гр. ІІІ-64		

Ми використовуємо термін перемикач OpenFlow для позначення перемикача, який приймає протокол OpenFlow. OpenFlow не є стандартом IETF. Натомість специфікація перемикача OpenFlow, центральний стандартний документ для OpenFlow, підтримується консорціумом OpenFlow: OpenFlowSwitch.org.

OpenFlow пропонує технологію віртуалізації мережі. Перемикач OpenFlow може бути налаштований для обробки одночасно як спеціалізованого мережевого трафіку (включаючи нестандартні експериментальні протоколи), так і звичайного мережевого трафіку. OpenFlow дозволяє безперешкодно співіснувати декільком типам трафіку. Ми побачимо, що наявність виробничої мережі має вирішальне значення для OpenFlow, оскільки виробнича мережа дозволяє контролеру спілкуватися з комутатором. Що ще важливіше, навіть якщо комутатор має звичайні правила переадресації, OpenFlow може встановлювати винятки. Наприклад, якщо звичайні правила переадресації направляють трафік для IP-адреси призначення X до заданого порту комутатора, OpenFlow дозволяє адміністратору вказати, що IP-трафік для X, який походить з джерела IP-адреси Y, повинен бути перенаправлений на інший порт комутатора.

					ДП 6430. 00.007 Д4			
Зм.	Арк	№ докум.	Підпис	Дата	Частина навчального курсу присвячена темі вступу в SDN.			
		Шапран А.С.						
Перевірив.		Долголенко О.М.						
Н. Контр.		Симоненко В.П.						
					Літ. Аркуш Аркушів 1 1			
					НТУУ “КПІ ім. Ігоря Сікорського” ФІОТ гр. ІІІ-64			

